DEPARTAMENTO DE INGENERÍA DE SISTEMAS Y AUTOMÁTICA
ESCUELA SUPERIOR DE INGENIEROS
UNIVERSIDAD DE SEVILLA

A Multi-Hypothesis approach for Range-only Simultaneous Localization and Mapping with Aerial Robots

por
Felipe Ramón Fabresse

PROPUESTA DE TESIS DOCTORAL
PARA LA OBTENCIÓN DEL TÍTULO DE
DOCTOR POR LA UNIVERSIDAD DE SEVILLA
Sevilla, Diciembre 2016

Directores:
Prof. Aníbal Ollero Baturone
Prof. Fernando Caballero Benítez

UNIVERSIDAD DE SEVILLA
Memoría presentada para optar al grado de Doctor por la Universidad de Sevilla

Autor: **Felipe Ramón Fabresse**
Título: **A Multi-Hypothesis approach for Range-only Simultaneous Localization and Mapping with Aerial Robots**
Departamento: **Departamento de Ingeniería de Sistemas y Automática**

V°B° Director: _____
Aníbal Ollero Baturone

V°B° Director: _____
Fernando Caballero Benítez

El autor: _____
Felipe Ramón Fabresse

A mi familia
A Delia

Agradecimientos

Los resultados de esta tesis doctoral han sido fruto de un trabajo duro y continuado, que no hubiera sido posible sin mis tutores, amigos y familiares. Quisiera agradecer a cada una de esas personas su apoyo y confianza no solo en los momentos más arduos del desarrollo de esta tesis, sino también en los buenos momentos compartidos y por ello me sentire eternamente en deuda con cada uno de ellos.

Me gustaría agradecer de forma más especial el apoyo y la confianza recibida por mis tutores Fernando Caballero y Aníbal Ollero al ofrecerme la oportunidad de formar parte del Grupo de Robótica, Visión y Control (GRVC) para el desarrollo de mi máster y tesis doctoral. Por una parte, quisiera agradecer a Aníbal Ollero su confianza a la hora de ofrecerme la posibilidad de comenzar a desarrollar mi carrera en el área de robótica con mi proyecto fin de carrera desde el Centro Avanzado de Tecnologías Aeroespaciales (CATEC) y su confianza a la hora de darme la oportunidad de continuar en el grupo de Robótica, Visión y Control de la Universidad de Sevilla para el desarrollo de mi máster y tesis doctoral. Por otro lado, quisiera agradecer de una manera muy especial a Fernando Caballero todos los conocimientos aportados a lo largo de todos estos años para el desarrollo de este proyecto y, sobre todo, por apoyarme en todos aquellos momentos en los que más lo necesitaba. Fernando ha supuesto ser para mi más que un profesor, un amigo con el cual he disfrutado de innumerables momentos y charlas que me han aportado un desarrollo a nivel profesional y personal.

Por otra parte, me gustaría agradecer el soporte ofrecido por el departamento de Ingeniería de Sistemas y Automática de la Escuela de Ingenieros de la Universidad de Sevilla, y al equipo de ingenieros e investigadores del Centro Avanzado de Tecnologías Aeroespaciales (CATEC) por permitir hacer uso de sus instalaciones y materiales para el desarrollo de los diferentes experimentos de esta tesis. También, me gustaría agradecer a mis compañeros de laboratorio el apoyo ofrecido a

lo largo de esta tesis. En especial a Jesús Martín, Victor Blanco, Arturo Torres, Alfredo Vázquez y Ricardo Ragel, así como a mis compañeros de despacho con los cuales he compartido muy buenos momentos a nivel profesional y personal. También quisiera agradecer al Institut de Robòtica i Informática industrial (IRI) de la Universidad Politécnica de Cataluña por darme la oportunidad de realizar una estancia en sus instalaciones para el desarrollo de experimentos con robots aéreos, en especial a Ángel Santamaría y Juan Andrade por su apoyo y conocimientos aportados durante dicha estancia.

Así mismo, me gustaría agradecer a Guillermo Perez por permitirme comenzar a formar parte de un equipo de ingenieros fantástico en Intel Corporation en los últimos años de desarrollo de esta tesis doctoral, sin que ello haya supuesto un obstáculo en la finalización de la misma o en mi desarrollo profesional en la compañía.

Finalmente, y no por ello menos importante, agradecer a mi familia, y en especial a mi novia Delia, el apoyo, cariño, confianza y paciencia ofrecidos a lo largo de estos años. Sin duda, sus consejos y apoyos han sido de gran motivación para la superación de los mejores y peores momentos afrontados a lo largo de esta tesis doctoral.

Esta tesis ha sido parcialmente financiada por el Ministerio de Economía, Industria y Competitividad (FPI: BES-2012-051840).

Resumen

Los sistemas de Range-only SLAM (o RO-SLAM) tienen como objetivo la construcción de un mapa formado por la posición de un conjunto de sensores de distancia y la localización simultánea del robot con respecto a dicho mapa, utilizando únicamente para ello medidas de distancia.

Los sensores de distancia son dispositivos capaces de medir la distancia relativa entre cada par de dispositivos. Estos sensores son especialmente interesantes para su application a vehículos aéreos debido a su reducido tamaño y peso. Además, estos dispositivos son capaces de operar en interiores o zonas con carencia de señal GPS y no requieren de una línea de visión directa entre cada par de dispositivos a diferencia de otros sensores como cámaras o sensores laser, permitiendo así obtener una lectura de datos continuada sin oclusiones.

Sin embargo, estos sensores presentan un modelo de observación no lineal con una deficiencia de rango debido a la carencia de información de orientación relativa entre cada par de sensores. Además, cuando se incrementa la dimensionalidad del problema de 2D a 3D para su aplicación a vehículos aéreos, el número de variables ocultas del modelo aumenta haciendo el problema más costoso computacionalmente especialmente ante implementaciones multi-hipótesis.

Esta tesis estudia y propone diferentes métodos que permitan la aplicación eficiente de estos sistemas RO-SLAM con vehículos terrestres o aéreos en entornos reales. Para ello se estudia la escalabilidad del sistema en relación al número de variables ocultas y el número de dispositivos a posicionar en el mapa. A diferencia de otros métodos descritos en la literatura de RO-SLAM, los algoritmos propuestos en esta tesis tienen en cuenta las correlaciones existentes entre cada par de dispositivos especialmente para la integración de medidas estÃąticas entre pares de sensores del mapa.

Además, esta tesis estudia el ruido y las medidas espúreas que puedan generar

los sensores de distancia para mejorar la robustez de los algoritmos propuestos con técnicas de detección y filtración. También se proponen métodos de integración de medidas de otros sensores como cámaras, altímetros o GPS para refinar las estimaciones realizadas por el sistema RO-SLAM. Otros capítulos estudian y proponen técnicas para la integración de los algoritmos RO-SLAM presentados a sistemas con múltiples robots, así como el uso de técnicas de percepción activa que permitan reducir la incertidumbre del sistema ante trayectorias con carencia de trilateración entre el robot y los sensores de destancia estáticos del mapa.

Todos los métodos propuestos han sido validados mediante simulaciones y experimentos con sistemas reales detallados en esta tesis. Además, todos los sistemas software implementados, así como los conjuntos de datos registrados durante la experimentación han sido publicados y documentados para su uso en la comunidad científica.

Acknowledgments

The results of PhD have been the outcome of a hard and continuous work, which would not have been possible without my advisors, friends and family. I would like to thanks each of them for their unconditional support and trust not only in the hardest moments along this thesis, but also for all great experiences shared with them. Is because of that, that I will always feel indebted with all of them for their help

I would like to thank in a special way the support and trust received from my advisors Fernando Caballero and Aníbal Ollero to offer me the opportunity to become part of their research group of Robotics, Vision and Control (GRVC) along the development of my master and PhD. On the one hand, I would like to thank to prof Aníbal Ollero to give me the opportunity to start the developments of my professional career on the robotics field in the Advanced Center of Aerospace Technologies (CATEC) y to allow me continue this career in the robotics, vision and control research group he leads at the University of Seville for my master and PhD. On the other hand, I would like to express an special gratitude to Fernando Caballero for all the knowledge transmitted along this thesis, and specially for its support when I most needed. Fernando became more than my advisor, a friend, with who I have shared amazing moments and talks that helped me to not only improve in my professional career but in my personal life.

Furthermore, I would like to thank to the team of engineers and researchers of the Systems and Automation Engineering department from the Engineer school at the University of Seville and of the Advanced Center of Aerospace Technologies (CATEC) to allow me to use their materials and facilities for the development of the different experiments carried out along this thesis. Also, I would like to thank to my laboratory mates, specially to Jesús Martín, Victor Blanco, Arturo Torres, Alfredo Vázquez and Ricardo Ragel, as well as my office mates with who

I have shared amazing professional and personal experiences. Additionally, I would like to thank to the Institut de Robòtica i Informática industrial (IRI) of the Politechnique University of Catalunya for giving me the opportunity to stay there to perform some aerial robot experiments, specially to Ángel Santamaría and Juan Andrade for their support and knowledge.

In the same way, I would like to thank to Guillermo Perez for giving me the chance to be part of an amazing engineering team at Intel Corporation in the last years of my PhD, which has not supposed any issue to end my PhD or to improve my professional career at the company.

Last, but not least, I would like to thank my family, and specially my girlfriend Delia, for their support, love, trust and patience along all years of development of this thesis. With no doubt, their advices and support have been of special motivation to get beyond the best and worst moments suffered in this PhD.

This thesis has been partially supported by the Spanish Ministry of Economy, Industry and Competitiveness (FPI: BES-2012-051840).

Abstract

Range-only Simultaneous Localization and Mapping (or RO-SLAM) aims to create a map composed of a set of range-only sensors with unknown initial position while at the same time it localizes the mobile robot with respect that map using just range-only observations.

Range-only sensors measure the distance between a pair of devices. They are specially suitable in aerial vehicles due to their small size and weight. Additionally, they can work indoor or in the absence of GPS signal and without requiring a direct line of sight between each pair of sensors as opposed to other classical sensors like cameras or LIDAR sensors, allowing a more continuous state observation without occlusions.

However, the lack of bearing information between each pair of range-only sensors leads to a rank-deficiency of the observation model where bearing parameters are the hidden variables to be estimated. When the dimensionality of the problem is increased from 2D to 3D environments for aerial applications, the number of hidden variables increase, making the problem more challenging in terms of computational efficiency, specially on multi-hypotheses implementations.

This thesis studies and proposes different approaches to make the application of RO-SLAM more efficient and feasible for real applications. This approaches take into account the scalability of the problem with respect the number of hidden variables and landmarks. As opposed to other methods proposed in the literature, the algorithms proposed in this thesis pay special attention to the particular inter-landmark correlations which appear in RO-SLAM, specially when introducing inter-landmark measurements.

Additionally, the thesis studies the noise and outliers of range-only sensors and proposes methods for their detection and avoidance. Data fusion techniques are also proposed for the integration of other sensor observations to refine the

estimations of the proposed RO-SLAM algorithm. Last chapters study and propose different techniques for the application of proposed RO-SLAM algorithms to multi-robot applications, as well as the use of active perception techniques to reduce the uncertainty of the system when the robot is following a bad trajectory with respect the trilateration of landmarks.

The proposed methods are validated with simulations and real experimentation detailed in this thesis. Furthermore, the software packages developed and the real datasets used during real experimentation have been documented and published to make it available to the research community.

Contents

List of Figures

List of Algorithms

List of Tables

Chapter 1

Introduction

1.1 Overview and motivation

One of the most important capabilities of an autonomous mobile robot is its self-localization within an environment with a certain degree of accuracy. Another common required task is to build a map of the surrounding environment (i.e. mapping problem). When both capabilities are required, then this combined task is known as Simultaneous Localization and Mapping (SLAM), which stands for the capability of a mobile robot for building a map of an initially unknown environment while at the same time it gets localized relative to this map.

Most common Unmanned Aerial Vehicle (UAV) localization systems are based on the combination of GPS with other inertial observations. However, GPS signal is not always available and inertial observations by their own introduce small errors which leads to a biased localization. Some examples with poor or zero visibility of the GPS signal include: warehouses, tunnels, disaster sites, etc. Then, other complementary sensors are required to allow robots be precisely localized in these environments. Figure 1.1 shows most common alternative sensors used for UAV localization task.

LiDAR sensors (in the literature also known as range sensors) use distance measurements between the mobile robot and different targets located around the robot with known relative orientation to give a precise localization of the robot when the targets position are known. LiDAR distance observations are measured illuminating a target with a laser in a known orientation and analyzing the reflected light or using a receiver sensor. The main drawback of this kind of sensors is their weight, so they are not suitable in many cases for micro UAVs which are usually very constrained with respect to their payload. In addition,

(a) (b) (c) (d)

Figure 1.1: Classic localization sensors: (a) 3D Li-DAR sensor, (b) Ultrasonic sensor, (c) Monocular camera and (d) Stereo camera

SLAM approaches based on 2D LIDAR suffers from inaccuracies associated to the use of a 2D sensor in 6DoF systems as is the case of aerial vehicles.

Ultrasonic sensors are a similar technology which measures the distance between the mobile robot and targets using an acoustic signal. However, it is usually based on directional emitters, so it is required to install an array of acoustic range sensors to measure distance at different orientations. Ultrasonic array sensors have significantly smaller resolution and frequency than LiDAR sensors, the reason why these sensors are more used as proximity sensors for obstacle avoidance than for mapping purposes.

Monocular and stereo cameras have become one of the most used sensors for aerial perception in the last decade due to their low payload, size, cost and the highly informative data they provide. Monocular cameras requires a triangulation of the targets to be localized since they lack of depth information (i.e. they are bearing-only sensors). In the case of stereo cameras, the needed stereo base-line to achieve a good depth resolution reduces the application of stereo cameras in UAVs. In order to tackle this issue, depth cameras offer a solution which is based on the use of a infrared projector and an infrared camera where the depth range depends on laser power.

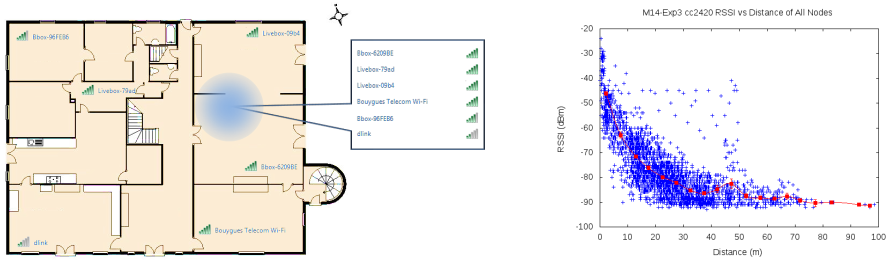
Despite all these devices are widely used nowadays, there are several situations where this kind of technologies cannot be used. For example, vision-based and laser-based sensors cannot be used when lighting conditions are not adequate, or even when there is not a direct line of sight (NLOS) between the camera and the target, or when there are partial occlusions of the object to be tracked.

Range-only sensors are able to measure just the distance between a pair of these devices or between the sensor and a target using reflection techniques. The main difference with respect LiDAR (or range) sensors, is that range-only devices provides distance information but without the relative orientation between the emitter and the receiver as is the case of LiDAR sensors. Since, sonar sensors or directional RFID antennas are included are based on oriented range observations, range-only sensors refers to omni-directional range observations which lack of bearing information. These sensors are specially interesting in Aerial Vehicles since they tend to be small and lightweight compared to some of the sensors presented above.

Range-only sensors can be found in the market implemented with different technologies like ultrasonic, laser-based, radio-based or even a combination of these technologies as shown in [85]. The most common range-only sensors in the market are radio-based sensors due to their ability to provide distance measurements without requiring a direct Line of Sight (LoS) between the emitter and a receiver, and because of the longer signal range they provide with respect cameras, LiDAR or ultrasonic sensors. Despite most radio-based technologies were based on ISM band (i.e. 2.4GHz band like WiFi), new devices uses Ultra-Wide Band (UWB) which make the technology more robust against reflections and obstacles like walls at expense of a reduced range of operation. To measure distance information with radio-based technologies one of the following methods are typically used:

- **Radio signal strength (RSS)**: these range-only sensors measure the distance between a pair of these devices by measuring the signal power [103, 107]. These are the most common sensors found on the market although they are the less accurate ones due to the imprecisions of the power to distance model. An example of these range-only sensors are the power measured by a WiFi receiver from different access points. In this situation the receiver is placed on the mobile robot which moves around in an environment that contains different WiFi access points like in museums, big malls, etc.

- **Time of Arrival (TOA)**: also known as Time of Flight (ToF) sensors. These sensors measures the distance between a pair of devices by measuring



(a) Example of Signal Strength received from different access points

(b) Signal strength vs distance

Figure 1.2: RSS range-only sensors

the time the signal delays to be received from the emitter or master to the receiver or slave [20, 53]. To get a better precision and in order to avoid to have a precise clock synchronization between emitter and receiver, manufacturers tends to use a derived technique called Two way ranging which consist on sending a signal from emitter to receiver and later return an ACK signal from receiver to emitter including the processing time taken by the receiver since the emitter signal was received and until the ACK is sent. Other improved methods over this technology have been presented in the literature. One example is the Nanotron Symmetric Double Sided Two Way Ranging (SDS-TWR), which will be detailed in Chapter 2. Other variants are presented in [2, 48, 61].

- **Time Differential Of Arrival (TDoA)**: Although this method is more focused on the localization of the emitter device, it is also considered in the literature as an additional ranging technology. This method is based on the a multilateration technique from an emitter to multiple receivers. The method consist on the intersection of two hiperbolas obtained by computing the difference in the time of arrival between the emitter and two fixed receivers, where the position of receivers are known [21, 105]. Then at least 3 receivers are needed in the case of 2D localization and 4 for 3D localization of the emitter.

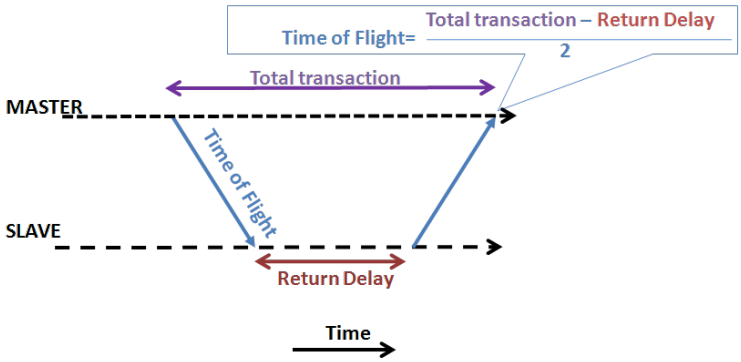


Figure 1.3: ToF range-only sensors

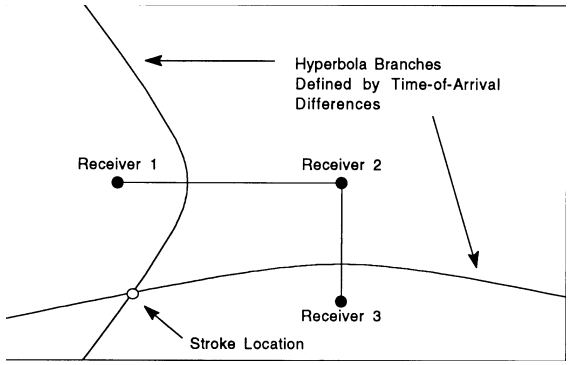


Figure 1.4: TDoA range-only sensors

For map building, there are different types of maps in the literature to be differentiated. In a first classification, maps can be differentiated between topological and metric maps. Topological maps are composed by qualitative features of the environment. A common example of topological maps are underground maps, which only gives information about connections between different underground stations but does not give any information about the distance between them or even their relative position. This dissertation is focused in metric maps, which are a kind of map that contains geometrical information about the environment in which the mobile robot moves. In metric maps, a second classification might be

done: grid-based vs landmark-based maps. Grid maps divides the environment in cells of fixed size. Each cell represents a free or non-free area where the mobile robot can move. These maps are common when using raw observations like those provided by LiDAR or sonar sensors based on signal reflection. The nature of this raw data make difficult to differentiate points of the environment from each other, but the relative position of the detected points from the current robot pose makes possible to build a grid map in which the robot can be localized. On the other hand, in landmark-based maps (also known as feature-based maps in Visual SLAM), each landmark represents a uniquely identifiable object. Each landmark contains a set of features which represent the position of the landmark in the map, but can also contain other kind of features like color, shape, etc, specially when using more informative sensors like cameras.

Another classification of SLAM approaches is made in the literature depending on the type of technique applied to solve the SLAM problem, like Graph-SLAM, EKF-SLAM, SEIF-SLAM, etc.

Depending on the type of sensor and map used for SLAM, different SLAM approaches can be found. Thus, for metric SLAM the most common SLAM approaches are Grid-SLAM and Visual-SLAM. The first is based on grid maps and LiDAR or sonar sensors, whereas the latest is based on landmark-based maps and monocular or stereo cameras.

Range-only SLAM (or RO-SLAM) is a metric SLAM that aims to create a landmark-based map while at the same time it localizes the mobile robot with respect that map using range-only observations. In this case, landmarks are the position of a set of fixed range-only sensors similar to the visual landmarks used in Visual-SLAM but with less informative observations (only distance between mobile robot and landmark).

Since radio-based range-only sensors provide a continuous estimation without having a direct line of sight between robot and target, this capability makes them very interesting in some applications for aerial and ground robots. They are specially suitable for disaster areas or low visibility UAV applications where cameras and other common sensors cannot operate due to a lack of visual texture. RO-SLAM is also interesting in indoor environments with reduced or non-existing GPS visibility.

However, the application of RO-SLAM with UAVs has become a challenging research area in the last decade due to the rank-deficiency of the range-only observation model. These observations are just composed by single value which

represents the distance between a pair of devices but does not contain the bearing information of the source signal (i.e. relative azimuth and elevation angle between sensing devices). This reduced observability requires using multi-hypotheses approaches to get at least an initial estimation of landmarks position.

Multi-hypotheses approaches tend to be computationally expensive and to scale poorly. Additionally, this computational burden becomes higher when increasing the dimensionality for Aerial Robotics applications in contrast to the opposite problem known as bearing-only SLAM, where the orientation between robot and landmark is known and the distance is the unobservable variable. In range-only SLAM, a higher dimensionality of the problem (e.g. from 2D to 3D SLAM) requires to estimate not only the azimuth relative angle between the robot and the target range-only sensor, but also the relative elevation angle. This dissertation focuses on the efficient, scalable and robust application of range-only multi-hypotheses methods for 3D Range-only SLAM for Unmanned Aerial Vehicles.

Landmark-based mapping problems usually require to solve the so called data association problem. Data association is the problem of uniquely identifying a landmark in order to be able to recognize it in successive observations of the same landmark (loop closures). This problem is typically found in Visual-SLAM where landmarks are composed of visual features and the system must be able to select the proper landmark features in order to differentiate landmarks from each other. The use of radio-based range-only sensors allow to eliminate the problem of data association since either, ultrasonic and radio signal, contains an identification code which allows to associate each measure to a unique pair of emitter and receiver devices.

Next sections reviews the state of the art of Range-only SLAM and last section introduces the main contributions of this dissertation and explains how the rest of the document is structured.

1.2 Related work

The current state of the art of SLAM is mature and many approaches have been proposed, including stable and long term algorithms for robot localization and mapping using stereo vision [82], RGBD [54], monocular vision [23], LIDAR [65] and other sensors. This is a well established scientific area and many solutions have been proposed for UAVs [14, 58, 70]. However, the constraints imposed by

some systems led to use non-line-of-sight (NLOS) sensors in order to account for partial or total occlusions of the landmarks, or as a complementary sensor, as will be shown in this dissertation.

Range-only methods have awakened a research interest in the last decade particularly for robot/people/object indoor localization and ubiquitous applications among others and more recently in aerial robotics like radio frequency source localization in military, rescue, aerial manipulation or inspection scenarios.

In the literature, it is possible to distinguish between three types of problems: one known as localization [71] and [6], where a mobile agent (e.g. a robot) is being localized with respect a set of landmarks (i.e. other range-only sensors). The second problem is known as network localization or mapping, where a set of static range-only sensors try to estimate the topology of the sensor network by measuring the distance (or range) between each pair of sensors or even between a mobile agent and each of the static sensors. The third problem would be a combination of previous ones. This problem is known as Simultaneous Localization and Mapping. For this problem, different solutions have been proposed for range-only sensors in the literature which will be analyzed in this section. A more focused analysis is given on each chapter, but here it is analyzed the general RO-SLAM problem.

The main research interest of range-only methods resides in how to cope with the multi-modality of the position distribution. Thus, in the case of range-only localization, several methods [51, 56, 71] are based on numerical optimization approaches which trilaterates the position of the mobile robot employing 3 or more static ranging nodes (also known as anchors) at different positions. On the other hand, [47] proposes a fingerprinting method which uses a neural network particularly useful when RSSI-based devices are used. Fuzzy logic has also been used for range-only localization [56] employing a Voronoi diagram to cope with common flip ambiguities in the probability distribution associated to range-only estimations. Other range-only localization approaches are based on Bayesian filters [46, 49, 51, 64] or batch-processing techniques [69].

For the mapping problem some early works use batch-processing techniques like Multidimensional Scaling (MDS) [60] or Least square methods [1] to map the relative position of each node. However MDS methods requires a high connectivity between static nodes to localize each node, the reason why other authors proposed other approaches based on sub-map estimation [76] or the use of artificial nodes created from a set of range measurements taken from different robot positions [30]. Other authors have used decentralized inference to solve

the mapping problem by means of multilateration from a mobile robot using probabilistic frameworks like particle filters [15, 16]. Particle filter models the inherent multi-modality of range-only landmarks position by using Monte Carlo sampling methods.

In the case of Gaussian filters, authors tends to use two common approaches: the first, and most common, consist on a delayed initialization of the Gaussian filter based on a pre-estimated position of landmarks [16, 80] and the second approach uses undelayed initialization based on multi-hypotheses frameworks to cope with the multi-modality of landmarks' position. However, in delayed initialization approaches, single estimation convergence will always depend on robot's trilateration with respect the landmark so that important delays may be produced until these landmarks converge and can be integrated in the Gaussian filter used to refine the robot position. On the other hand, undelayed approaches have the advantage of integrating range-only observations into the Gaussian filter since the very beginning without lost of information and, more important, they are able to improve the robot's position estimation without requiring single solution convergence of landmarks. One of this undelayed approaches [27, 29] is based on a polar parametrization which allows to initialize the Gaussian filter using a predefined variance around the $\rho\theta$-space. The main drawback of this approach is the use of heuristics based on the robot trajectory to split the initial unimodal distribution into two Gaussians which, in the case of 3D RO-SLAM, becomes more complex. In [17] it is proposed a method which integrates a Gaussian Mixture in an Extended Kalman Filter (EKF) to represent the multi-modality of the bearing distribution, this approach has the additional advantage of making possible the integration of inter-node range-only observations without loosing cross correlation information between landmarks as is the case of the decentralized approach presented in [27]. The main drawback of multi-hypotheses methods is the computational burden used to keep all possible hypothesis in the system. To cope with this drawback, [17] uses a prune strategy which allows to reduce the computational burden of the multi-hypotheses approach as landmarks converge to a single solution.

For the complete RO-SLAM problem, different solutions have been proposed. Thus, in [10] a novel solution is presented using a spectral SLAM algorithm which estimates the mobile robot and landmarks position using a singular value decomposition (SVD) of the observation matrix. The main advantage is a significant reduction of the required computational burden using classical methods.

However, this method is oriented to batch processing, making difficult its implementation for online estimation applications. Other authors use probabilistic frameworks which are suitable for online estimation and can model most belief distributions present in robotic applications [92]. Most common solutions for RO-SAM are based on a set of SLAM frameworks which are known as EKF-SLAM, UKF-SLAM, FatSLAM and others. A comparison of these frameworks is done in [66, 68]. In [66] it is shown how the unscented FastSLAM presents better results over other classical methods based on EKF or UKF. However, FastSLAM solutions don't preserve the correlation between different landmarks of the map in those applications in which it might exist. Thus, for example, in [9, 104, 106], a FastSLAM solution is proposed using a particle filter for robot localization and another one for each landmark estimation (i.e. for each radio emitter). In [106] an optimization on landmarks particle filter is proposed reducing the number of particles and hence decreasing the computational burden of the filter. On the other hand, [104] optimizes the problem using an adaptive re-sampling method which dynamically reduces the number of particles required for each landmark. Other landmark-based SLAM algorithm is considered in [52], where the authors use a particle filter to initialize the EKF filter for each new landmark of the FastSLAM. The main drawback of the last solution is the use of a delayed initialization, so that first measurements are not integrated into the EKF, instead the Kalman filter is initialized when the particle filter has converged. This is an important factor in SLAM because the localization estimation convergence will significantly depend on landmarks initialization, so that a delayed initialization of landmarks might cause a loss of information for the mobile robot position parameters.

The use of a Gaussian-mixture model allows to integrate the multi-modality of the observation model for each new landmark, integrating all received measurements since the very first time. The use of a Gaussian-mixture to model the multi-modality of the observation model is used in [8] to solve the RO-SLAM for 3D state spaces. Here, a FastSLAM framework is also used, but landmarks are initialized using multiple hypotheses. Each hypotheses is a Gaussian with different pose parameters and all these Gaussians has a related weight which compose a complete GMM. To update these Gaussians, the authors employs a single EKF for each landmark hypotheses. Despite the efficiency of the solution in [8], this approach is not suitable to model the high inter-landmark correlation which appears in RO-SLAM when integrating inter-landmark observations (i.e. distance measurements between landmarks and not only from robot to landmark). These

kind of applications is one of the main motivations of this dissertation. Thus, this dissertation is focused on the improvement of state-of-the-art multi-hypotheses approaches for centralized filters like EKF-SLAM in order to make feasible to use them to model inter-landmark correlations.

1.3 Outline and main contributions

This dissertation is divided in 8 chapters and two appendices. Here, a summary of these chapters is presented:

- ***Chapter 2* - 3D RO-SLAM with delayed initialization**

  RO-SLAM poses serious challenges mainly related with its low-informative measurements (distance between two sensing elements) that leads to multiple localization hypotheses that do not fit well with the usual linear/gaussian approximations, the reason why researchers had researched different approximations to solve this problem. As a first approach, this chapter presents a common non-parametric Bayesian filter for simultaneous localization and mapping problem. The approach detailed in this chapter is based on a Monte Carlo algorithm, commonly known as Particle Filter. The filter is used to get a first estimation of landmarks position (i.e. delayed initialization of the landmark). The vehicle localization is always estimated with an Extended Kalman Filter (EKF) which is dynamically increased with those landmarks whose particle filter has converged into a single Gaussian distribution. Finally, the complete SLAM problem is carried out as an EKF-SLAM. The chapter provides a set of results for localization, mapping and the complete SLAM problem showing how the quality of the initialization of landmarks in a centralized Kalman filter might affect not only the localization of the robot but also the mapping of others landmarks.

- ***Chapter 3* - Efficient undelayed 3D RO-SLAM based on multi-hypotheses**

  In order to integrate all information since the very first measurement received, and hence improve the estimation of the vehicle position, this chapter presents an undelayed approach for RO-SLAM employing a multiple-hypothesis solution and a centralized EKF which will be explained in detail. The chapter presents a reduced parametrization to cope with the common poor scalability of multi-hypotheses solutions. Additionally, this chapter

shows different approaches to update the weights of the hypotheses and new observation models which, in contrast to Federated Information Sharing approach used in [17, 89], does not require to split the information of range-only observations between all hypotheses since it uses a single equation to correct all hypotheses. The chapter shows how these efficient observation models can also be used with other state parameterizations already present in the literature. Experimental results with a real aerial vehicle and a set of range-only sensors show the feasibility of the approach, comparing the results with other well known undelayed RO-SLAM methods. The contents of this chapter have been published in two conference papers [34, 35], in a workshop [42] and in a journal article which is under review [40].

- ***Chapter 4*** **- Towards robust 3D RO-SLAM estimation**

  Range measurements are subject to different sources of errors as BIAS or scaling due to walls or sensor aging. This section extends the 3D RO-SLAM with online estimation of the propagation model for every range-only sensor involved in the estimation. In addition, an outlier rejection method is described in order to filter out erroneous range measurements that may worsen the RO-SLAM estimation. The contents of this chapter has been published in one conference paper [37] and in a journal article [38].

- ***Chapter 5*** **- Integrating 3D RO-SLAM and visual markers**

  As introduced above, one of the main advantages of radio-based range-only sensors is their ability to work without a direct LOS between sensors and their large range of operation. This makes these devices specially useful to reduce the exploration time in those applications which have to map the position of a set of elements to be manipulated or inspected. However, visual-based algorithms might help to refine the mapping precision once the landmark/object has an initial position estimation from the range-only technology. For those applications, this chapter presents an approach of visual and range-only observations fusion, showing how visual measurements can complement range-only estimations by refining them. Other sensors like the altimeter are used to help to reduce the altitude uncertainty of the robot position. The contents of this chapter has been published in a conference paper [36]. This chapter is also comming from a stay in the Institut de Robótica i Informática industrial (IRI) at the Technical Univer-

sity of Catalonia (UPC) for the integration of the visual marker detector developed at IRI with the algorithms developed in this thesis.

- ***Chapter 6*** **- Cooperative 3D RO-SLAM**

  This chapter describes the approach implemented to reduce the local belief of the map fusing the map information coming from other cooperative aerial robots. In this chapter it is described how this multi-SLAM approach can be implemented to fuse this information even when the landmarks to be fused are still in a multi-hypotheses situation. The ability to integrate other robots information from landmarks which are in multi-hypotheses stage allows to quickly reduce the number of hypotheses for a better convergence of the filter. The chapter shows how this multi-SLAM approach not only fasts the convergence of the map but also improves indirectly the local robot position estimation. The contents of this chapter has been published in a conference paper [43] and in a journal article [45].

- ***Chapter 7*** **- Active perception for 3D RO-SLAM**

  In this chapter it is described an approach used to move the robot in such a way it gains information from landmarks and hence can reduce its uncertainty faster and avoid common flip ambiguities. To do so, the approach is based on the maximization of a function with the gain of information. In this chapter, two different behaviors are considered: path following and exploration behavior. To implement this combination of goals the method uses a navigation architecture from the literature known as Distributed Architecture for Mobile Navigation (DAMN). The contents of this chapter has been published in a conference paper [39] and will be sent to the Journal of Intelligent and Robotics Systems.

- ***Chapter 8*** **- Conclusions and future work**

  To summarize the methods proposed and the results obtained, this section is aimed to make some conclusions about the main strengths and weakness of the approaches presented in this work to solve the range-only SLAM problem in three-dimensional spaces. Later a brief description of some future improvements will be described in order to improve the methods proposed.

- ***Appendix A*** **- Introduction to probabilistic filtering**

As part of this thesis, a general purpose library called Open Probabilistic Filtering (OPF) has been developed and released for educational, research or other purposes. This chapter will introduce the basis of probabilistic filtering techniques implemented on this library. First, some probabilistic theory will be introduced ending with Bayes rules. Bayes probabilistic theory is then used to explain the basis of bayesian filters for parametric, non-parametric and semi-parametric algorithms.

- ***Appendix B* - Aerial Robot Datasets using Range-only Sensors**

  This appendix describes the details of the different datasets recorded and used for real experimentation along this thesis. The appendix also gives the links to the webpage created for the datasets, and the link to ROS wiki where the drivers of the sensors developed in this thesis have been published. The contents of this appendix have been sent to the Journal of Field Robotics and is pending to review process.

The main objective of this dissertation is to develop new RO-SLAM techniques for three-dimensional spaces for either, indoor and outdoor environments. The techniques studied in this dissertation will take into account the stochastic nature of the aerial vehicle dynamics and observation models.

Selecting the appropriate SLAM framework requires taking into account the application where the strategies to be implemented will be applied. For example, if the map landmarks are highly correlated in the application considered, a centralized solution should be applied in order to take into account this correlation but, in general terms, a decentralized solution would scale better for most applications so that it will be necessary to try to decentralize the solution as much as possible.

Summarizing, the main contributions of this thesis are:

1. Development of efficient 3D RO-SLAM techniques.
2. Improvement on the robustness of the RO-SLAM approach using pre-filtering methods of range-only measurements and efficient integration of inter-landmark observations in centralized filters.
3. Improvement on the scalability of multi-hypotheses RO-SLAM approaches.
4. Development of fusion techniques with other sensors like altimeters and visual-based target detection and localization.

5. Implementation of multi-hypotheses solutions proposed in a multi-SLAM application with multiple aerial vehicles.

6. Implementation of an active perception technique for mapping uncertainty reduction.

7. Published datasets for indoor/outdoor real experimentation with aerial robots for public research use [41].

8. Development of two full documented libraries, one for general purpose probabilistic Bayesian filtering and a second library with RO-SLAM approaches developed in this thesis. These libraries will be published as Open Source software for sharing with research community. Additionally an executable has been linked with these libraries to integrate this software with the Robotics Operating System (ROS) framework [86]. This ROS package will be published along with other ROS packages developed for the monitorization of the algorithm results.

9. Development and publication of full documented range-only drivers and data messages developed for radio-based range-only sensors used for real experimentation in ROS community [32, 33, 44].

This dissertation also provides the validation of the proposed methods employing not only simulation results but also real datasets from experiments based on real projects that support this thesis.

All these contributions have been published in the following articles.

- International conference articles:
  - [35] FABRESSE, F. R., CABALLERO, F., MAZA, I., AND OLLERO, A. Undelayed 3d RO-SLAM based on gaussian-mixture and reduced spherical parametrization. In 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (Tokyo Big Sight, Tokyo, Japan, Nov. 2013), pp. 1555-1561.
  - [34] FABRESSE, F., CABALLERO, F., MAZA, I., AND OLLERO, A. Localización de vehículos aéreos basada en 3D RO-SLAM con inicialización no retardada empleando mezcla de Gaussianas. In ROBOT 2013: FIRST IBERIAN ROBOTICS CONFERENCE (Centre for Automation and Robotics (CAR), Madrid, Spain, Nov. 2013).

  - [37] FABRESSE, F. R., CABALLERO, F., MAZA, I., AND OLLERO, A. Robust range-only SLAM for aerial vehicles. In 2014 International Conference on Unmanned Aircraft Systems (ICUAS) (May 2014), pp. 750-755.
  - [36] FABRESSE, F., CABALLERO, F., MAZA, I., AND OLLERO , A. Localization and mapping for aerial manipulation based on range-only measurements and visual markers. In 2014 IEEE International Conference on Robotics and Automation (ICRA) (May 2014), pp. 2100-2106.
  - [43] FABRESSE, F., CABALLERO, F., AND OLLERO, A. Decentralized simultaneous localization and mapping for multiple aerial vehicles using range-only sensors. In 2015 IEEE International Conference on Robotics and Automation (ICRA) (May 2015), pp. 6408-6414.
  - [39] FABRESSE, F.R., CABALLERO, F., MERINO, L., AND OLLERO, A. Active perception for 3D range-only simultaneous localization and mapping with UAVs. In 2016 International Conference on Unmanned Aircraft Systems (ICUAS) (2016).
  - [83] PEREZ-GRAU, F. J., FABRESSE, FELIPE R. CABALLERO, F., VIGURIA, A., AND OLLERO, A. Long-term aerial robot localization based on visual odometry and radio-based ranging. In 2016 International Conference on Unmanned Aircraft Systems (ICUAS) (2016).

- International journals:

  - [45] FERNÁNDEZ, J. C., MARTINEZ-DE DIOS, J.R., MAZA, I., FABRESSE, F.R., AND OLLERO, .A. Ten Years of Cooperation Between Mobile Robots and Sensor Networks.
  - [38] FABRESSE, F.R., CABALLERO, F., MAZA, I., AND OLLERO, A. Robust range-only SLAM for unmanned aerial systems. Robotics Journal of Intelligent and Robotic Systems (2015).
  - [40] FABRESSE, F., CABALLERO, F., AND OLLERO, A. An efficient approach for undelayed range-only SLAM based on gaussian mixtures expectation. Robotics, IEEE Transactions on. (Submitted)

  - [41] FABRESSE, F. R., CABALLERO, F., AND OLLERO, A. Indoor and outdoor aerial robot datasets using range-only sensors. `http://grvc.us.es/staff/caba/roslam`. Journal of Field Robotics (Submitted)

- Released software:

  - [33] Range-only driver stack for ROS framework (`http://wiki.ros.org/rangeonly_driver`). The stack includes the following packages:
    * [44] Nanotron radio-based range-only driver package for ROS framework (`http://wiki.ros.org/nanotron_swarm`)
    * [32] General purpose range-only sensor data message package for ROS framework (`http://wiki.ros.org/rangeonly_msgs`)

- Workshops:

  - [42] FABRESSE, F., CABALLERO, F., AND OLLERO, A. Novel multi-hypotheses approach for 3D range-only SLAM using aerial vehicles. In ICRA Workshop on AERIAL ROBOTS PHYSICALLY INTERACTING WITH THE ENVIRONMENT (2014).

1.4 Thesis framework

One of the main objectives of this thesis was to implement and validate the different methods developed employing unmanned aerial vehicles. The experiments were carried out with simulations and real data sets which have later been published for public research use. The contents of the thesis has been focused on three European projects and two national projects, all of them described below. The purpose of these experiments is to validate the algorithms developed in those cases where other common technologies cannot be applied.

- **Aerial Robotics Cooperative Assembly System (ARCAS):**

Figure 1.5: Installation of a platform for evacuation of people

Figure 1.6: Installation of landing platforms

This is an European project from the Seventh Programme Framework which concerns the design and development of the first free-flying robot system in the world for cooperative assembly. The project will pave the way for a large number of applications including the building of platforms for the evacuation of people in rescue operations (Figure 1.5), the installation of platforms in uneven terrains for landing of manned and unmanned VTOL aircrafts (Figure 1.6), the cooperative inspection and maintenance (Figure 1.7) and the construction of structures (Figure 1.8).

Cooperative transportation could be applied to deploy mobile robotic systems on structures that are not directly accessible by some mobile robotic



Figure 1.7: Other cooperative assembly applications



Figure 1.8: Structure assembly by means of aerial robots

system (see Figure 1.9 as an example). Here the application of aerial robots could be exploited to transport and locate modular robotic systems. The transportation of parts and remote assembly could be used to avoid the payload limitations of aerial robots. The assembly functionality of the aerial robot could also be useful to replace parts such as, for example, the batteries of a mobile robotic system mechanically designed to simplify the replacement operation.

In these applications it is very important to have an accurate localization system which tracks the vehicles position specially for cooperative tasks. On the other hand, the different elements to be manipulated cooperatively by multiple vehicles are not always in the line of sight of these vehicles and hence need to be located with sensors which allows to perceive elements occluded by buildings, trees or other objects of the environment, so an efficient and accurate mapping of elements to be manipulated would help

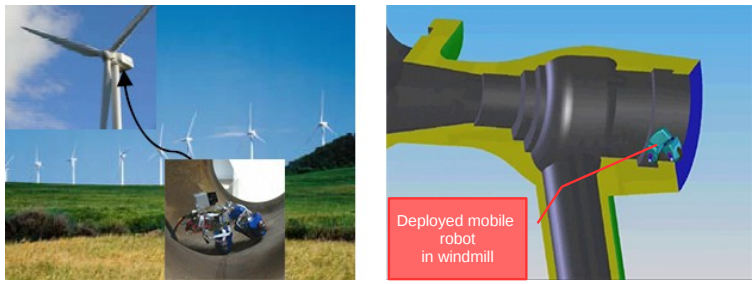


Figure 1.9: Aerial inspection robots

vehicles to manipulate them. The algorithms developed in this thesis dale with the efficiency of the mapping solution for the case of 3D mapping. Also, in this project, as range-only sensors are integrated in the elements to be manipulated, in most of the cases one element might contain more than one range-only sensor. The integration of range-only sensors on the same element imposes high correlations on the relative position of these devices (i.e. landmarks of our mapping problem) which needs to be modeled and hence inter-landmarks observations and correlation become specially important in this application.

This project also requires precise perception techniques for the assembly of structural elements, hence data fusion from different sensors becomes specially important in this application to refine the estimations from Range-only SLAM module. In that sense, this dissertation presents fusion techniques by means of Extended Kalman Filtering between range-only, visual and other navigation sensors.

- **Estimation And Control For Safe Wireless High Mobility Cooperative Industrial Systems (EC-SAFEMOBIL):**

  This is also an European project from the Seventh Programme Framework, devoted to the development of sufficiently accurate common motion estimation and control methods and technologies in order to reach levels of reliability and safety to facilitate unmanned vehicle deployment in a broad range of applications. It also includes the development of a secure architecture and the middleware to support the implementation. Two different kind of applications are included in the project:

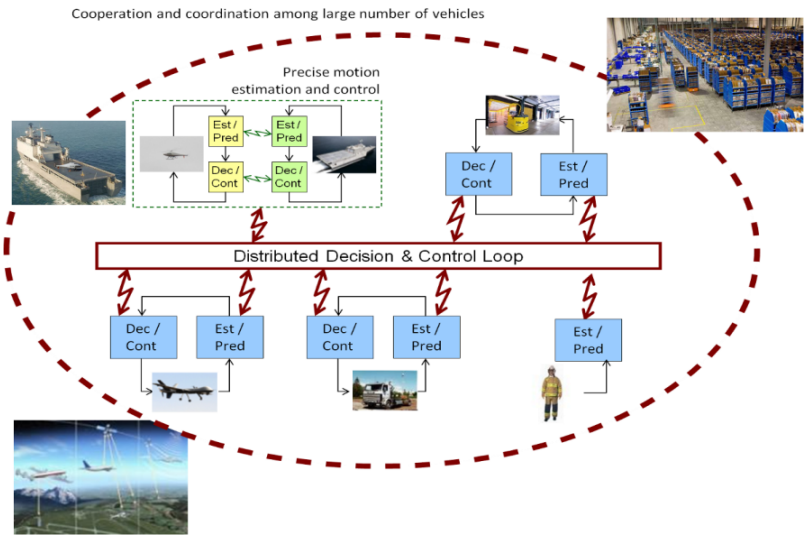


Figure 1.10: EC-SAFEMOBIL decision and control architecture

- Very accurate coupled motion control of two mobile entities. The technologies will be demonstrated in two challenging applications dealing with the landing on mobile platforms and launching of unmanned aerial vehicles from a manned vehicle.
- Distributed safe reliable cooperation and coordination of many high mobility entities. The aim is to precisely control hundreds of entities efficiently and reliably and to certify developed techniques to support the exploitation of unmanned platforms in non-restricted areas. This development will be validated in two scenarios: industrial warehousing involving a large number of autonomous vehicles and surveillance also involving many mobile entities.

The motivation of EC-SAFEMOBIL arises from two observations: 1) The important synergies among high mobility applications in different sectors related to multi-billion EUR businesses such as transportation, warehousing, surveillance, security and many others; 2) distributed estimation and

cooperative control methods are scarcely applied in high mobility industrial systems because of lack of reliability, safety and security.

Lack of reliability is usually related to uncertainties in the models and estimations and unexpected events, but is also dependent on hardware, wireless communications and architectural issues. Safety plays a critical role in the coordination and cooperation of high mobility systems. Many examples can be found in industrial applications such as transportation, warehousing, logistics and others (see Figure 1.11). The methods to optimize these systems should also consider safety constraints to avoid collisions between mobile entities and with static obstacles. The above mentioned replacement of complex mechanical devices by networked embedded systems, or the enhancement of the mechanisms by means of these embedded systems, could increase safety.

Security is also a requirement for practical applications of high mobility networked systems. Without security mechanisms in place, any malicious subject could easily affect the integrity and availability of both the application layer and the underlying network infrastructure endangering the high mobility system.

For this project, this thesis has contributed on the development of cooperative perception methods for RO-SLAM. Additionally, other requirements like robustness to dale with the uncertainty in the observation and dynamics model are taken into account, as well as techniques for outliers rejection. As stated before, the thesis focuses mainly in efficiency and robustness in order to properly track each vehicle individually and to map the position of different range-only sensors embedded in elements of the environment (docking and landing places, objects to be monitored, etc).

- **Cooperative Long Endurance Missions With Aerial Robots (CLEAR):**

  This is a national research project which also supported my research grant during the development of this thesis. The main objective of this project is the research, development and implementation of a system with cooperative aerial and ground robots for the performance of large endurance missions

(a) In-flight docking and landing systems

(b) Traffic management

(c) Vehicles tracking and filming crowds

Figure 1.11: EC-SAFEMOBIL applications

involving automatic recharging of batteries or refueling like described in Figure 1.12.

New cooperation methods involving aerial robots are applied to maintain long endurance missions in indoor and outdoor scenarios. This objective includes the development of new methods for range only positioning and simultaneous localization and mapping (SLAM) and methods for the planning of the cooperative missions.

Hence, one of the objectives of this project is the research in new methods for range-only SLAM combined with topological and cognitive algorithms. Particularly, application of Gaussian Mixture approaches for range-only SLAM based on radio-signals were applied for localization and mapping. This SLAM framework allows the integration of other position information,

(a) Laser based battery charging

(b) Cooperative refueling

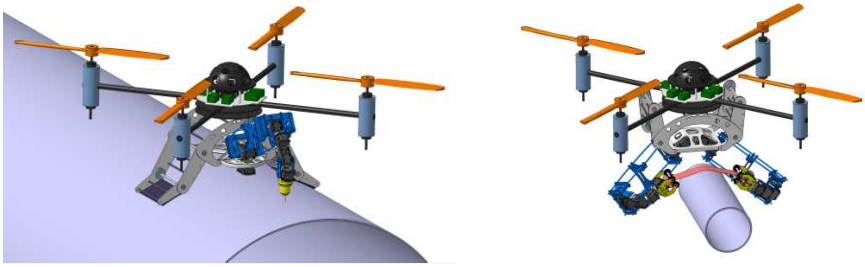
Figure 1.12: CLEAR applications

as vision-based or GPS, for localization and map refinement. Previous knowledge of the environment and its properties will also be used to achieve robustness against disturbances such as mobile or changing objects and improve functions such as data association and loop closure detection.

Also new methods for distributed robust estimation combining the measurements obtained from on-board sensors and the information from the communication system will be researched. Then it will be possible to decrease the uncertainties due to inaccuracies of the sensors. Here, distributed methods will be used, that scale well with the number of robots and objects being observed.

- **AErial RObotic system integrating multiple ARMS and advanced manipulation capabilities for inspection and maintenance (AEROARMS):**

  AEROARMS is an European project from Horizon 2020 Program Framework which extends the developments of ARCAS project for aerial manipulation using multiple arms. The project poses complex mechatronics, control, telemanipulation, perception and planning problems. Particularly, new mechatronic designs and control methods are needed to implement the aerial robots with several arms for grabbing and flying operations. Advanced control strategies with force and vision feedback are also required for manipulation. Also new coordinated control strategies of the multiple arms and the aerial platforms are needed. Moreover, new perception methods, with ability to adapt to changing illumination conditions, are required for accurate local mapping and localization in denied GPS industrial en-



(a) Docking and drilling task (b) Multi-arm free flying operations

Figure 1.13: AEROARMS applications

vironment, grabbing and manipulation, and integration of sensory data to help the operation of the crawler. On the other hand, planning is required to compute the required motions of the aerial robot for efficient and safe grabbing and manipulation taking into account the constraints imposed by the aerial robot and the environment, including motion planning for torque compensation, and for cooperation with the ground mobile robot. Moreover, reactive behaviors to increase safety, avoiding collisions with objects in the environment will be also implemented.

The developments of this thesis will allow this project to overcome the problem of Simultaneous Localization and Mapping under GPS denied and lack of visual features conditions, providing a more continuous SLAM estimation from radio based range sensors and the RO-SLAM approaches presented in this thesis. Robust estimation will be critical in this project when working on industrial scenarios where multipath effects might affect the RO-SLAM filter with the introduction of multiple outliers.

- **AErial RObotic manipulation system for the MAINtenance in the energy generation and distribution with application to wind generators (AEROMAIN):**

  AEROMAIN is a national project which also supported my research grant during the development of this thesis. This project proposes the develop-

ment of the worldwide first aerial robotic system with advanced manipulation capabilities to be applied in inspection and maintenance of energy systems and particularly in the maintenance of wind turbines, including contact inspection (i.e. ultrasonic inspection) and blade repairing of surface damages or even "materials lost" of impacted areas (leading edge). These operations are today very costly and performed manually in risky conditions with experts that have to climb, using for example a rigging system, to inspect specific areas and to perform repairing. The application of crawler robots with magnetic wheels does not solve the problem because the blades of many new wind turbines are made of composites and the crawler should be deployed and maintained.

The application of the methods developed in this thesis are going to be applied for the mapping of the crawlers to be deployed on top of the turbines and for the precise localization of the aerial platform using the mapped position of multiple range-only sensors deployed around the working area.

Chapter 2

3D RO-SLAM with delayed initialization

2.1 Introduction

As a first approach, a non-parametric solution for initialization of landmarks of RO-SLAM has been implemented during the first developments of this thesis in order to be able to compare the results of delayed and undelayed initialization results.

The chapter describes the details of how to implement a common non-parametric Bayesian filter for the initialization of landmarks and how can this non-parametric distribution be integrated in a Gaussian filter once the landmark position converges to a Gaussian distribution. Thus, for simultaneous localization and mapping (SLAM) problem, this chapter implements a delayed initialization EKF-SLAM approach. The vehicle localization is estimated with an Extended Kalman Filter (EKF) which is dynamically increased with those nodes whose particle filter has converged into a single Gaussian distribution. Thus, at the end, the complete SLAM problem is carried out as a centralized EKF-SLAM which is able to deal with inter-landmark correlations.

2.2 Overview and related work

This chapter proposes a RO-SLAM approximation which extends the mapping solution presented in [16], but adding the localization of the robot for the complete SLAM problem. The proposed SLAM solution is mainly based on an EKF-SLAM which initially contains only the estimation of the aerial vehicle position. On the other hand, a set of range-only sensors are used as the main sensor to correct the estimation of the robot localization. However, the location of these sensors (or landmarks) is completely unknown and hence, their position should also be estimated by means of mapping algorithms.

To map the position of these sensors with an EKF, it is necessary to have an initial position estimation of them. Due to the low-informative measurements provided by these devices, (i.e. only distance between robot and one landmark), it is necessary to employ an initialization strategy which allows to get an initial estimation of landmarks which deals with the complex observation model of such kind of sensors. In this thesis, it is considered the use of range-only sensors such as radio emitters or ultrasonic sensors which provides the identification of each landmark then, data association problem is directly solved by using the unique identifiers provided by range-only sensors.

For delayed landmark initialization, several approaches have been proposed in the literature. Thus, for example, the method proposed in this chapter is based on the mapping approach detailed in [16], where landmarks are initialized using a particle filter and later switched to a Gaussian filter by computing the Gaussian parameters from the particle filter PDF. The method is similar to the one proposed in [15, 93], but in these later papers an information filter is employed to get a decentralized estimation of the map allowing a more scalable solution without traking of the beacon inter correlation information. In the last approach all beacons are initialized with a PF on the Unmanned Aerial Vehicle (UAV) which is localized using a GPS. Later, once each beacon converges, its estimation is refined in a local beacon information filter making use of the computational resources each node of a Wireless Network Sensor.

Other methods propose improvements over the PF, reducing the amount of particles and hence the resources required by the method [104, 106]. Also, some improvements are proposed taking advantage of Wireless Network Sensor computational capabilities in [94, 95, 97]. However, this SLAM methods are based on 2D RO-SLAM. For 3D RO-SLAM, [9] proposed a FastSLAM approach, where each beacon is initialized in a particle filter which later is switched in an EKF included in the map of the FastSLAM framework. However, although [9] used a 3D mapping approach, the approach was implemented for ground robots. A 3D RO-SLAM with aerial vehicles is implemented in [98], where the author uses a supervisor in order to efficiently introduce inter-landmark measurements during the initialization stage of the landmarks particle filter to reduce the convergence time of the individual particle filters.

The method presented in this chapter proposes a delayed initialization based on a sequential Monte Carlo filter (or Particle filter) which, as stated above, consist on sampling the state space of the sensor location using an initial probability

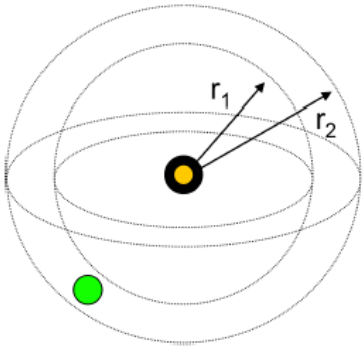


Figure 2.1: Initial uniform distribution of a range-only sensor position when only one range measurement is received from an aerial robot. The estimated position from which the robot detected the landmark the first time (i.e. first range-only observation received) is depicted with a yellow point and represent the center of the uniform spherical distribution in which the landmark might be located. The real position of the landmark is depicted with a green point.

distribution.

In this case, the initialization stage takes place when the first range measurement r_i is received from beacon i. This stage consist on initializing a new particle filter P_i^t which particles are drawn according to the probability distribution of the range observation model, like the one depicted in Figure 2.1. Then, once the particle filter converges into a single Gaussian distribution, the particle set P_i^t can be integrated in the same EKF where the robot localization takes place. By following this procedure with every landmark detected, the EKF vector state increase dynamically. At the end, the complete SLAM problem is solved in a centralized EKF.

2.3 Robot localization approach

In the SLAM strategy presented in this section, the localization of the vehicle is directly performed by the EKF, so that the first parameters of the state vector are the vehicle parameters. Thus, an initial value of these parameters must be defined, indicating the associated variance of the initial robot position in the covariance matrix of the EKF.

As the vehicle moves, new motion observations $\boldsymbol{u}^t$ are received. These observations are integrated in the EKF during the prediction stage according to the motion model of the robot to get a prior distribution of the new vehicle position. On the other hand, the correction stage of this localization problem consist on integrating the range measurements between the vehicle and the different landmarks detected. However, this correction stage can only be applied for those landmarks already integrated in the EKF state, i.e. once landmarks are included in the EKF state from delayed initialization strategy. This delay on the integration of landmarks into the EKF implies that the system will not integrate enough information for the correction on the robot 3D position until at least 4 landmarks are integrated in the EKF. This might cause some divergences on the vehicle localization estimation, making the EKF numerically unstable. For this reason, when employing delayed initialization methods, it is mandatory the use of anchors, i.e. range-only sensors which position is already known, so that the robot position estimation can be corrected with the measures of this anchors.

To get a good trilateration from anchors, they should be deployed throughout the environment taking into account the area coverage by each transmitter. At least 3 anchors are required to perform a good trilateration of a landmark when a 2D representation is considered, and at least 4 anchors for 3D representations.

2.4 Hybrid Mapping approach

This section explains how to setup and update the particle filters used during the initialization stage of each landmark, and how to determine the convergence of the particle filter so that, finally, these particle filters can be integrated into a centralized extended Kalman filter trading off between the precision of this estimation and the convergence velocity.

2.4.1 Landmark initialization

The particle filter is initiated when the first range measurement from the robot is received. The first range measurement defines the radio of the uniform annulus sphere distribution in which the real position of the landmark is located. The center of this sphere is directly related with the estimated position of the robot at this time $\boldsymbol{x}_r^t$. This uniform distribution is depicted in Figure 2.1, in this picture two radio values are depicted, r_1 and r_2, they represent the thickness of the spherical

shell, which depends on the deviation $\sigma(r_i)$. Particles are then initiated as 3D points represented in Cartesian coordinates $\boldsymbol{x} = [xyz]^T$ and distributed according to that spherical shell uniform distribution.

2.4.2 Particle filter: Update stage

The information about the state is updated with the set of measurements $\boldsymbol{z}^{1:t}$ received up to time t. This set of measurements consists of range-only observations r_i generated between sensor i and the aerial robot.

The information about the state during the initialization stage is represented by the conditional probability distribution $p(\boldsymbol{x}^t|\boldsymbol{z}^{1:t})$. This distribution (the posterior) can be estimated online while new measurements are received. Indeed, the position will be estimated and updated recursively.

In that sense, the likelihood function $p(\boldsymbol{z}^t|\boldsymbol{x}^t) = p(\boldsymbol{r}_i^t|\boldsymbol{x}^t)$ plays a very important role in the estimation process. In this case, this function expresses the probability of obtaining a given range value r_i from a sensor i given the position of the robot $\boldsymbol{x}_r^t$.

The model used here considers that, for each particle n, $p(\boldsymbol{r}_i^t|\boldsymbol{x}_n^t)$ follows a Gaussian distribution centered on the estimated distance $r_{i,n}^t$ with a standard deviation proportional to the standard deviation of the range-only sensor measurements.

$$r_i = \mathcal{N}(\mu(r_{i,n}^t), \sigma(r_i^t)) \tag{2.1}$$

Regarding the range-only sensor observation model, it is considered that these devices has an omnidirectional antenna with an isotropic propagation model. Thus the observation model of a single range measurement in the case of 3D representations (see Figure 2.2) is as follows:

$$r_i = \sqrt{(x_b - x_i)^2 + (y_b - y_i)^2 + (z_b - z_i)^2} \tag{2.2}$$

It can be seen how the value of the beacon position $[x_b, y_b, z_b]^t$ cannot be known with a single range measurement, so a trilateration process is required. In the case of 3D environments, trilateration requires to take range measurements from at least 4 different positions as Figure 2.3 shows. In Figure 2.3a, it is depicted how when the second range measurement (grey sphere) is received, the previous region of uncertainty (wired sphere) intercepts with the second one creating a circle of possible positions, where the range-only sensor (green sphere) might be located.

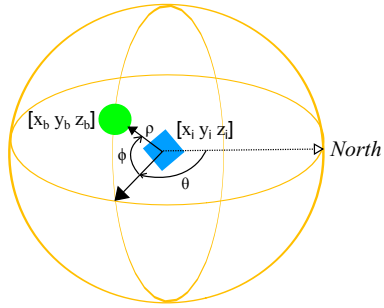


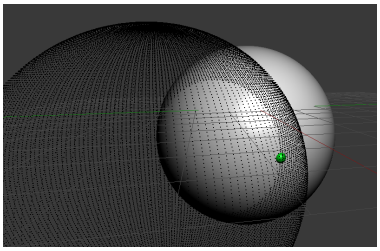
Figure 2.2: Observation model of a range-only sensor with omnidirectional antenna and isotropic propagation model. The green dot represents the receiver node and the blue rhombus represents the emitter node which requested the ranging measurement. the orange circles represent the possible positions of the receiver node when receiving only one range measurement.

Then, the problem can be seen as the trilateration problem in a 2D environment, when at least two new range measurements are required to precisely localize the range-only sensor. This situation is depicted in Figure 2.3b considering only the plane of the circle of possible solutions generated with the second range measurement. In Figure 2.3b the orange circle represents the circular distribution generated in the second measurement, the red circle represents the intersection of the third range measurement with the previous distribution and the blue circle represents the last range measurement. As it can be noticed, in order to have a good trilateration of the target, it is required to move with different values of coordinates x, y, z for each position, otherwise the distribution will generate what is known as flip ambiguities (i.e. multiple modes in the probability distribution).

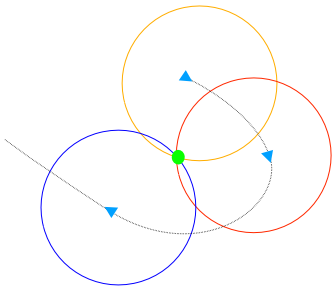
Once the parameters of the Gaussian observation model have been identified, the weights of the particles are updated considering the likelihood of the received data and the current likelihood of each particle (lines 1 to 5 of Alg. 1).

In detail, the update procedure is as follows: for each particle $\boldsymbol{p}_n^t$, the distance $r_n^t = ||\boldsymbol{p}_n^t - \boldsymbol{x}_r^t||$ is obtained. From this distance, the mean and variance of the conditional distribution $p(r^t|\boldsymbol{p}_n^t)$ are obtained, so that $p(r_i^t|\boldsymbol{p}_n^t) = \mathcal{N}(r_i^t; r_{i,n}^t, \sigma(r_i^t))$.

The probability of the actual range value under this distribution is finally employed to update the weight of the particle ω_n^t as:



(a) Intersection of two spherical shell distrib.



(b) Next annular distrib. intersections

Figure 2.3: 3D trilateration process: The green dot represents the real position of the landmark and the blue triangle represents the positions of the aerial robot at three different situations.

Algorithm 1: Particle filter for mapping range-only sensors

Input: P^{t-1} and r_i^t
Output: P^t

for $n = 1$ *to* N **do**
    sample $\boldsymbol{p}_n^t \sim p(\boldsymbol{x}^t|\boldsymbol{p}_n^{t-1})$
    Compute $r_{i,n}^t = ||\boldsymbol{p}_n^t - \boldsymbol{x}_r^t||$
    Determine $\mu(r_{i,n}^t)$
    Update weight of particle n as
    $\omega_n^t = \omega_n^{t-1} p(r_i^t|\boldsymbol{p}_n^t) = \omega_n^{t-1} \mathcal{N}(r_i^t; r_{i,n}^t, \sigma(r_i^t))$
Normalize weights ω_n^t
Compute N_{eff}
if $N_{eff} < N_{th}$ **then**
    Resample with replacement N particles from P^t.

$$\omega_n^t = \frac{\omega_n^{t-1}}{\sigma(r_i^t)\sqrt{2\pi}} e^{-\frac{(r_i^t - r_{i,n}^t)^2}{2\sigma(r_i^t)^2}} \tag{2.3}$$

When the filter is running, the weights of the particles with high likelihood increase, while most of the particles rest at places of very low likelihood on the state space.

As the number of particles is limited, a resampling algorithm (line 9 of Alg. 1) is included to compute the posterior probability $bel(\boldsymbol{x}^t)$. The algorithm duplicates particles with high weights and eliminates those with very low weights according to the probability $p(r_i^t|\boldsymbol{p}_n^t)$, i.e. according to the previously updated weights. It is very important to normalize the weights updated before resampling particles. The resampling method employed in this thesis is known as low-variance resampling method, which is described in Algorithm 2. This algorithm allows to spread the particles over the maximum likelihood areas. However, as Algorithm 2 shows, this resampling method differs from common low-variance resampling described in [92] in that it draws with a normal distribution centered in one of the most likely particles and with variance σ_{lvr} instead of just repeating the most probable particles,

Algorithm 2: Low variance resampling

Input: $\bar{P}^t$
Output: P^t

1 $P^t = \emptyset$;
2 $r = rand([0, N^{-1}])$;
3 $c = \bar{\omega}_1^t$;
4 $i = 1$;
5 **for** $n = 1$ *to* N **do**
6 $u = r + (n-1)N^{-1}$;
7 **while** $u > c$ **do**
8 $i = i + 1$;
9 $c = c + \bar{\omega}_i^t$;
10 draw $\boldsymbol{p}^t \rightarrow P^t$ according to $\mathcal{N}(\bar{\boldsymbol{p}}_i^t, \sigma_{lvr})$ with $\omega^t = \bar{\omega}_i^t$;
11 Normalize weights ω_n^t;

In order to overcome some of the known problems of resampling stage, an additional consideration is taken into account: resampling only takes place when the effective number of particles N_{eff} is below a threshold N_{th}. The effective number is computed as follows:

$$N_{eff} = \left[\sum_{n=1}^{N} (\omega_n^t)^2 \right]^{-1} \tag{2.4}$$

2.4.3 Switching from PF to EKF

As new measurements are received, due to the Gaussian nature of range-only observations, particles distribution tends to converge into a Gaussian distribution. In order to determine the convergence of the filter, two different approximations have been implemented in this thesis.

The first algorithm developed uses the averaged standard deviation σ_{P^t} as a measurement of convergence of the particle filter into a Gaussian distribution. The mean $\boldsymbol{\mu}^t$ and variance $\boldsymbol{\Sigma}^t$ of the Gaussian distribution can be computed from the actual particle filter state as:

$$\boldsymbol{\mu}^t = \sum_{n=1}^{N} \boldsymbol{p}_n^t \omega_n^t \tag{2.5}$$

$$\boldsymbol{\Sigma}^t = \sum_{n=1}^{N} (\boldsymbol{p}_n^t - \boldsymbol{\mu}^t)^2 \omega_n^t \tag{2.6}$$

Then, if the value σ_{P^t} is lower than a certain threshold σ_{th}, the particle set is considered to be following a distribution similar to the Gaussian distribution defined by parameters $\boldsymbol{\mu}^t$ and $\boldsymbol{\Sigma}^t$.

Another convergence criteria tested in this dissertation was the Kullback Leibler divergence factor. The Kullback Leibler divergence factor compares two distributions and gives a value of divergence which is near to 0 when there is a high similarity between both distribution. This divergence factor is used here to compare the Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}^t, \boldsymbol{\Sigma}^t)$, with the real distribution of the particle set, using the following equation:

$$KL = \int \omega_n^t ln(\frac{\omega_n^t}{\mathcal{N}(\boldsymbol{p}_n^t; \boldsymbol{\mu}^t, \boldsymbol{\Sigma}^t)}) d\boldsymbol{p}^t \tag{2.7}$$

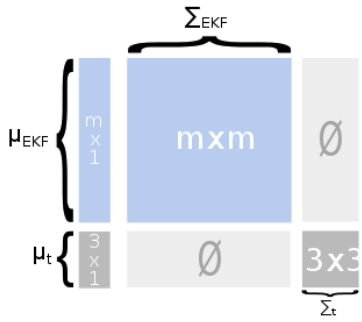


Figure 2.4: This figure shows how the state vector of a EKF can be extended with $\boldsymbol{\mu}^t$ and $\boldsymbol{\Sigma}^t$ extracted from a particle filter. The initial state vector of the EKF is has m parameters before incorporating the 3x1 mean vector from the particle filter.

But this convergence criteria was finally discarded due to the computational time required to compute this divergence factor for large number of particles.

When the PF has converged into a Gaussian distribution, the parameters $\mathcal{N}(\boldsymbol{\mu}^t, \boldsymbol{\Sigma}^t)$ can be integrated in the extended Kalman filter extending the current state vector. The procedure to include this Gaussian distribution into the current state vector of the centralized EKF is similar to the method used in [89]. The mean value is appended at the end of the state vector and the covariance matrix obtained from the particle filter is also included at the end of the EKF covariance filter, making the correlations with other elements of the actual EKF equal to zero. Figure 2.4 shows the procedure with a simple schema assuming that $\boldsymbol{\mu}^t$ is a column vector of 3 elements X, Y, Z and that the actual size of the EKF state vector is m.

Once a particle filter has been integrated in the centralized EKF, new range-only observations from the initialized sensor i are incorporated in this EKF according to the observation model (2.2).

2.5 Simulated and real experiments

To validate the SLAM framework proposed in this chapter, different experiments where carried out. The experiments where performed in a simulated environment and with a real data set. The range-only sensor used for real experiments throughout this thesis, and hence the one considered in simulations, is composed by a

radio emitter and a set of receivers. These radio beacons are able to measure the distance between each pair of devices by processing the ToF of the transmitted signal.

Next section characterizes the radio-based range-only sensor used during the experiments and then some simulation and real experiment results are presented.

2.5.1 Characterization of Nanotron radio-based range-only sensors

The range-only sensor employed for the experiments of this work consists on a set of radio beacons specially designed for Real Time Localization Systems (RTLS). The radio-based range-only sensor model employed is nanoPAN 5375 DK (see Figure 2.5), developed by Nanotron. This model has the following characteristics:

- ATMega 1284P microcontroller at 20MHz.
- Radio transceiver 2.4 GHz ISM band. Up to 20dB transmission power.
- Ranging accuracy of 2m indoors and 1m outdoors.
- Ranging frequency: 80Hz.
- 128KB flash memory for programs and retrieved data.
- Distance measurements computed with SDS-TWR technique without needing any clock synchronization between nodes.

These radio beacons implement a special communication protocol derived from the 802.15.4 standard called 802.15.4a. This protocol extends the low consumption characteristics from its parent standard with a physic layer specially designed for RTLS. Despite the protocol reduces the bandwidth for data transmission, it improves the physic layer to reduce interferences with other devices working on the ISM band (2.4GHz) by adding two channels specially designed for the modulation technique called Chirp Spread Spectrum (CSS). This modulation technique reduces some interference with other existing frequencies by working on two new channels and by employing a robust modulation (CSS modulation) which reduces the transmission errors with respect the modulation used in 802.15.4.

On the other hand, these radio beacons implement a ToF based technique which does not require a synchronization between sensors clock at expense of a

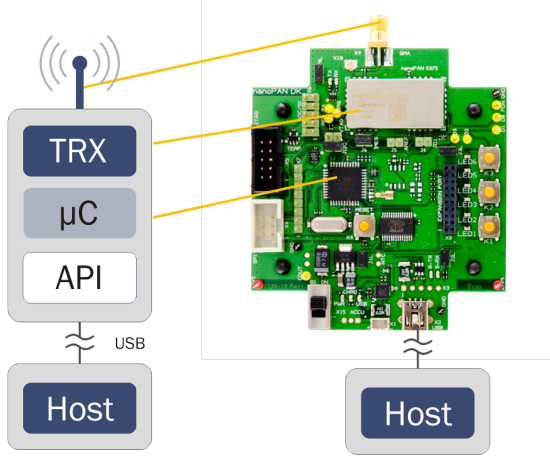


Figure 2.5: Range-only sensors employed during experiments: nanoPAN 5375 DB. The figure shows the essential parts of the nanoPAN 5375 radio beacon architecture. The API layer represents the drivers provided by the manufacturer to program the microcontroller to start ranging with other devices, etc. The last layer represents the host controller, i.e. the computer in which range measurements are processed.

reduced measurement frequency. The ToF technique used by nanoPAN 5375 DK is SDS-TWR introduced in Chapter 1, which works as follows (see Figure 2.6 for a graphical representation):

1. Emitter node (node A) sends a range request message to node B and saves the local time stamp t_{r1}.
2. Receiver node (node B) process the request message and sends an acknowledgement message to node A adding information about the time t_{p1} used to process the received request message.
3. Node A saves the time stamp t_{ack1} in which received the ACK message.
4. Steps 1, 2 and 3 are repeated starting from node B to avoid errors coming from different clock accuracies between Node A and Node B.
5. Node B sends the results t_{r2}, t_{ack2} and t_{p2} of the second ranging to node A.

Once the process has finished, the node which started the ranging cycle computes the distance between both radio beacons as follows:

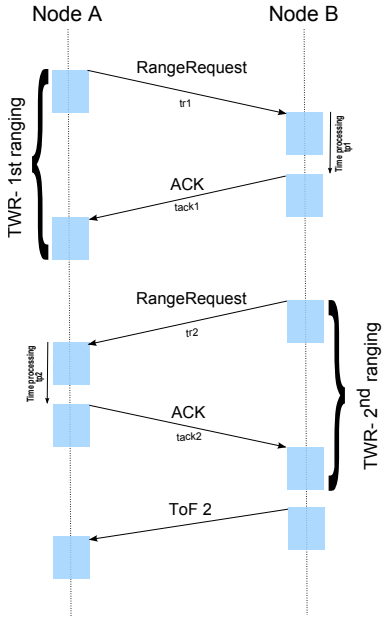


Figure 2.6: Symmetric Double-Sided, Two Way Ranging (SDS-TWR): method to measure the distance between a pair of radio-based devices employing ToF measurements. The method performs the operation in both directions to correct errors related to imprecisions on local clocks accuracy between emitter and receiver. Thus, SDS-TWR is an asynchronous method, i.e. it does not need any synchronization between emitter and receiver to compute the ToF.

$$r_{a,b} = \frac{(t_{ack1} - t_{r1} - t_{p1}) + (t_{ack2} - t_{r2} - t_{p2})}{2} \tag{2.8}$$

To get an approximated sensor model, first experiments performed during this dissertation were focused on characterizing the noise of range measurements when the emitter and the receiver are both static and placed at different distances. The setup employed for this experiment is shown in Figure 2.7, as this figure shows, experiments were carried out in indoor and outdoor environments, using a measuring tape to measure the real distance between both nodes. Different parameters of the sensors were configured until reaching the ones that gave us better results.

(a) Outdoor setup (b) Indoor setup (c) Host connected to base node (emitter)

Figure 2.7: Setup of nanoPAN 5375 characterization for outdoor and indoor environments.

Using these setups, the characterization of the range-only sensor was performed with two static nodes separated by 1, 2, 4, 8, ..., 30 meters. The results are shown in Figure 2.8, where green dots represent the received range measurements and red dots are the mean value at each distance tested, cyan and red lines represents the associated standard deviation of real measurements and mean values respectively. The results show a deviation of about 25cm for both, indoor and outdoor environments, which is quite better than the precision specified by the manufacturer (1m and 2m respectively).

Finally, a second characterization was performed employing the same configuration as for static characterization. In this case, the experiment was performed in order to test the effect of other phenomenon which appears when one of the radio beacons moves, like doppler shifting and others. Most of these effects are virtually suppressed by using the CSS modulation of 802.15.4a protocol but, as the results shown in Figure 2.9, the motion of a beacon still affects the signal propagation, and hence the ranging precision. The characterization experiments with motion where carried out with a mobile robot with a beacon on-board.

In this last experiment the ground-truth of the robot was computed with a precise Monte Carlo Localization (MCL) algorithm using laser data and a map of the environment in which the robot was moving through. In this case, the results in Figure 2.9 show a non constant behavior of the standard deviation (cyan line) with respect the mean value (green line) of samples registered (blue dots) from the moving robot to 5 different static nodes.

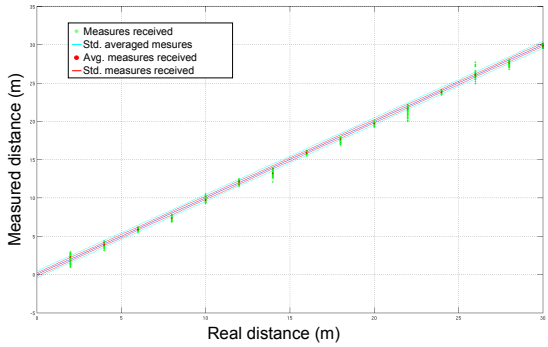


Figure 2.8: Characterization of nanoPAN 5375 with static beacons. X axis represent the real distance between beacons and Y axis represent the measured distance. Green dots represent the range measures received at each distance tested, and the cyan line represents the associated standard deviation of all measurements received. Red points represent the mean value of the range measures, while the red line represents the associated standard deviation of all mean values.

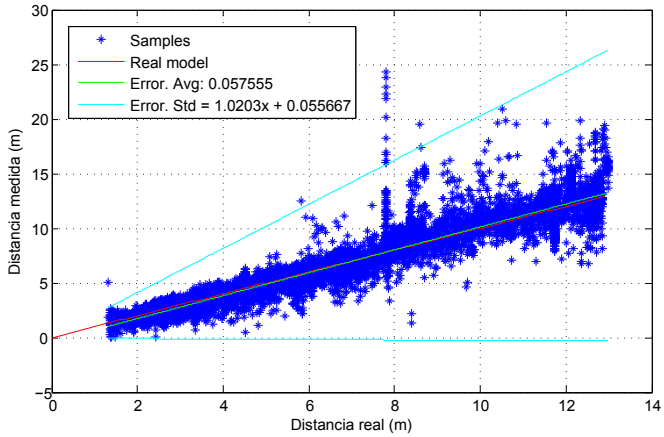
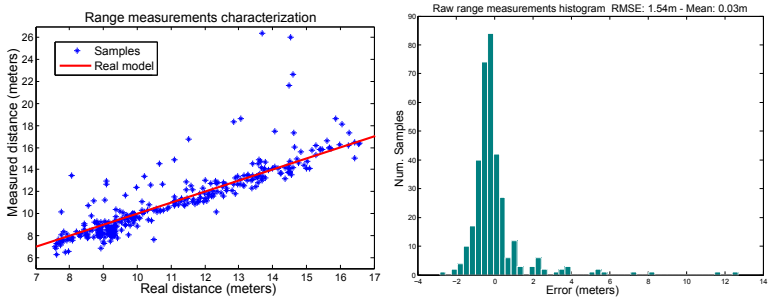


Figure 2.9: Characterization of nanoPAN 5375 with one beacon moving. X axis represent the real distance between beacons and Y axis represent the measured distance. Blue dots represent the range measures received while robot was moving. The green line represents the regression function of the mean values of range measurements received, while the cyan line represents the associated standard deviation of all mean values.



(a) (b)

Figure 2.10: Nanotron range-only sensor characterization for indoor environments with aerial robots: (a) sensor characterization (b) error histogram.

With these experiments, the characterized standard deviation follows a linear function (2.9) which increase with the distance between radio beacons.

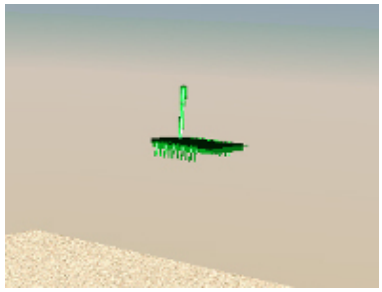
$$\sigma(r_i) = 1.025 r_i + 0.025 \tag{2.9}$$

This characterization was performed for indoor environments and with a ground robot.

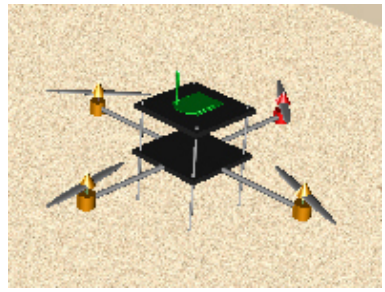
For aerial robots this linear increase of the standard deviation was not observed as shown in Figure 2.10. Instead, a fixed standard deviation of about 1.5m was characterized with some outliers as shown in the observations histogram of Figure 2.10b. This is due to a decreased multipath effect when the robot is flying. In this last experiment the ground-truth of the aerial robot was estimated using a VICON tracking system with millimetric positional precision. In this experiments, 5 static nodes were deployed throughout the environment and the emitter was embedded in the aerial platform. Similar results were obtained for outdoor characterization with the same aerial robot and using a differential GPS as ground-truth estimator with 10cm of precision. The setup of these indoor/outdoor real experiments will be detailed in Appendix B.

2.5.2 Simulations

For this first implementation of RO-SLAM, a Matlab prototype was developed. This section will test in simulation the delayed initialization approximation described in this chapter.



(a) Radio beacon range-only sensor



(b) Quadrotor with range-only sensor

Figure 2.11: Simulation experiments models the dynamics of an aerial robot (Quadrotor) and a set of radio beacons with the same sensors characterization obteined during real experimentation.

For this purpose, a simulator has been developed, which simulates not only the motion model of the vehicle but also the observation model of range-only sensors based on the static characteristic detailed in previous section for indoor environments (i.e. noise error of about 1.5m). A GPS was also simulated, in order to show the accuracy of the localization filter when fusing range-only sensors with other common navigation sensors.

Matlab Virtualization toolbox has been used (see Figure 2.11) for the visualization of the simulated environment, while other elements of the algorithms are plotted with common functions of Matlab. These visualization tools are used in this document to show the results of the different algorithms.

The first simulation considers the case where there is only one beacon to be mapped which real position is the center of the robot circular trajectory (Figure 2.12) and with a sinusoidal trajectory on Z axis. This trajectory was selected to avoid the lack of beacon state observability and have a good trilateration of the beacon position. In this first simulation, 4 anchors are used as the unique sensors to correct the estimated position of the aerial robot by the EKF while the beacon position is not initialized by the PF used.

The results of this experiment are shown in Figure 2.12, where the real position of the beacon and anchors are represented with a blue diamond an the real position and trajectory of the robot is depicted with a red cross (the red line is the trajectory until the current timestamp). The estimated position of the robot is represented with a light blue cross (the light blue line represents the estimated

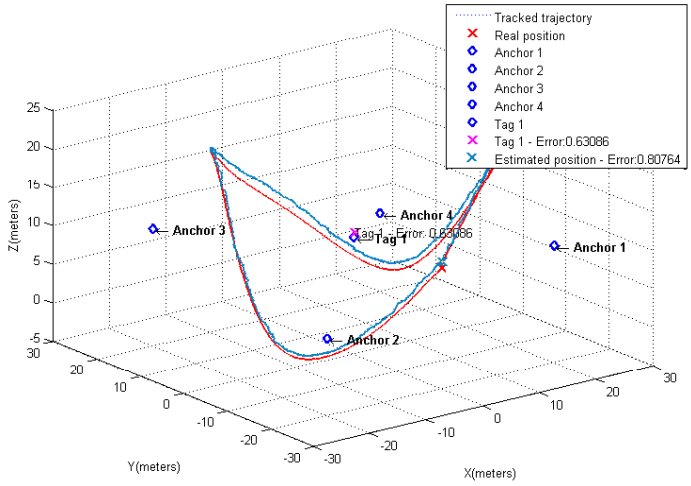
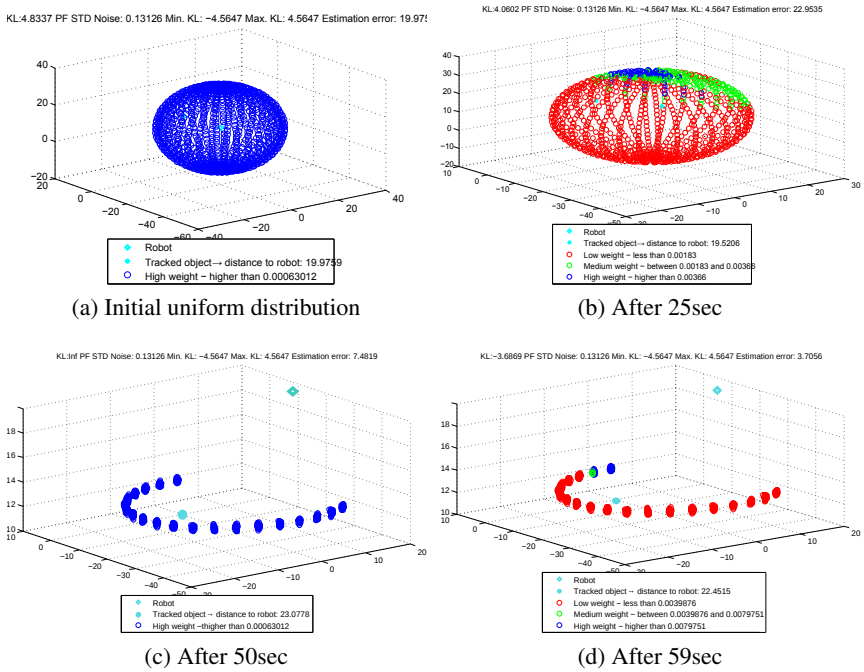


Figure 2.12: This figure shows the final results of a delayed RO-SLAM employing a particle filter to get a initial estimation of the beacon position for the EKF. In this experiments 4 anchors are used as the unique sensor to correct the estimated position of the aerial robot while the beacon has not been initialized yet.

trajectory until the current timestamp). The estimated position of the beacon is represented with a pink cross, this estimation corresponds to the estimation made by the EKF once the beacon has been initialized by the PF.

For the PF, the standard deviation σ_{lvr} used during the resampling algorithm of all simulations is 0.1 meters. This value was selected empirically after several simulations, being $\sigma_{lvr} = 0.1$ the best value which makes the resampled particles not diverge so much from the most probable areas. An example of convergence of a particle filter is shown in Figure 2.13. In this figure the cyan diamond represents the current position of the robot, while the cyan dot represents the real position of the beacon to be initialized. The circles represent the particle set P^t, where blue circles are the most probable particles (particles with higher weight), the green circles represent those particles with a probability between the most probables and the less probable particles, this particles are represented with red circles. The concentration of particles in different areas is caused by the resampling algorithm, where the sparse distribution around an area is the result of employing the standard deviation σ_{lvr} when drawing the new particles instead of replacing an existing



(a) Initial uniform distribution

(b) After 25sec

(c) After 50sec

(d) After 59sec

Figure 2.13: Example of particle filter convergence. The cyan diamond represents the current position of the robot, while the cyan dot represents the real position of the beacon to be initialized. The circles represent the particle set P^t, where blue circles are the most probable particles (particles with higher weight), the green circles represent those particles with a probability between the most probables and the less probable particles, which are represented with red circles.

particle from the prior distribution.

The covariance convergence threshold σ_{th} used is 3 meters, which is the same threshold used in [16]. Figure 2.14 represents the initial estimation of a beacon in the EKF, once the PF has converged. After several simulations the convergence delay with this robot trajectory is near to 1 minute (around 30 meters).

As Figure 2.14 shows, beacons are not always properly initialized (in this example the initial estimation error is near to 4 meters). As a consequence of this bad initialization the EKF estimations of the beacon does not converge and might diverge the estimation of the vehicle position.

The second simulation experiment was aimed to see the effect of fusing other

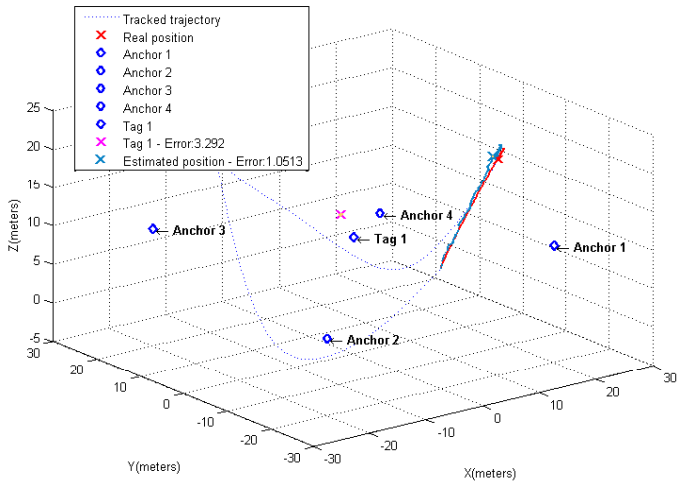


Figure 2.14: Example of the convergence of a PF into a Gaussian distribution. The picture shows the initial estimation of the EKF once the PF has converged. After several simulations the convergence delay for this trajectory and the velocity of the simulated aerial robot is near to 1 minute (around 30 meters).

sensor data into the EKF. Thus, a simple GSP simulator was developed with an horizontal error of about 2.5m and a vertical error of about 3m. The results of the delayed RO-SLAM solution fusing GPS information are shown in Figure 2.15.

The results show how the inclusion of new sensor smooths the estimation of the robot trajectory. On the other hand, the estimation of the beacon position is less accurate compared with previous results, but this is due to a bad initialization of the beacon estimation with the PF (near to 5 meters of initial error).

The last simulation experiment was focused on checking the effect of removing the anchor nodes from the filter. The results of this experiment are shown in Figure 2.16.

The final results show the estimation of the robot position is less precise. This is mainly related to the error associated with the GPS sensor (near to 3 meters). In this case, the estimation of the beacon position is quite better due to a better initialization of the EKF after the convergence of the PF.

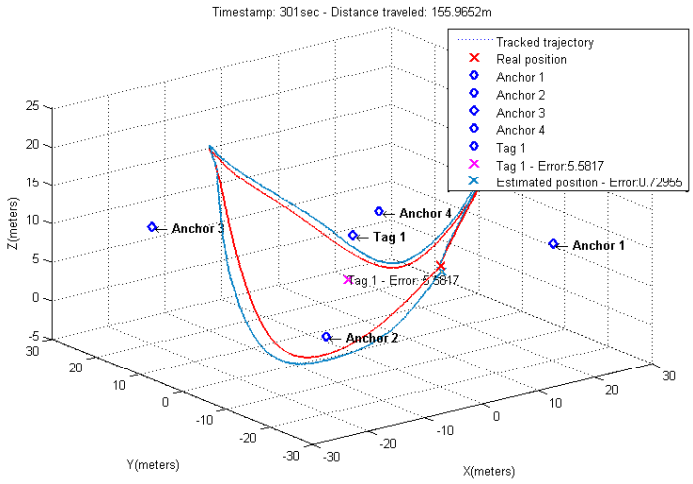


Figure 2.15: This figure shows the final results of a delayed RO-SLAM employing a particle filter to get a initial estimation of the beacon position for the EKF. In this experiments 4 anchors are used to correct the estimated position of the aerial robot together with a GPS sensor.

2.5.3 Real dataset

For real experiments, a robot was used together with the Nanotron network composed by 4 anchor nodes, 1 beacon node and 1 base node (node attached to the robot). In this case, the state of the robot is compound by its 2D position x_r, y_r and its azimuth angle θ_r. Although, this experiments estimates the 2D position of the robot, the position of the beacon is estimated in 3D coordinates.

The experiments were performed in the CONET testbed at the Engineering School (University of Seville) [72]. Figure 2.17 shows the setup used for this experiment.

For this experiment, a classic differential steering odometry model was implemented taking into account the current robot linear and steering velocity. The positions estimations based only on this odometry are referred as dead-reckoning estimations. The motion model used for dead-reckoning estimations is as follows:

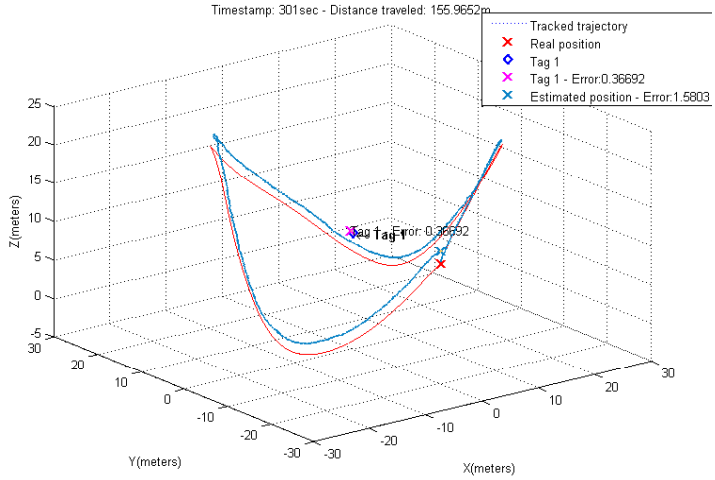
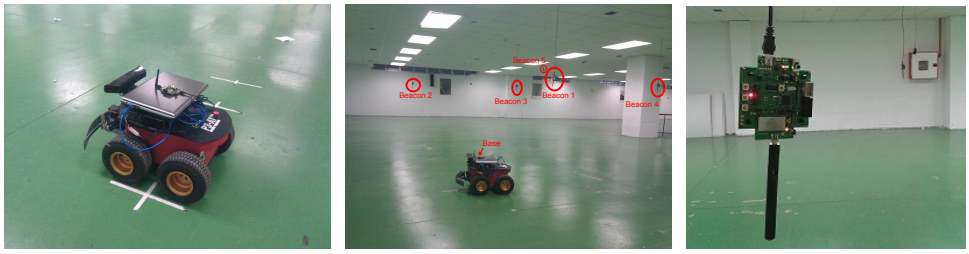


Figure 2.16: This figure shows the final results of a delayed RO-SLAM employing a particle filter to get a initial estimation of the beacon position for the EKF. In this experiments only the GPS is used to correct the estimated position of the aerial robot until the beacon converges to an initial EKF position.



(a) Pioneer 3-AT (b) Setup of radio beacons (c) Nanotron radio beacon

Figure 2.17: Setup used in CONET testbed for real experiments. Pioneer 3-AT ground robot was used together with the Nanotron network composed by 4 anchor nodes, 1 beacon node and 1 base node (node attached to the robot).

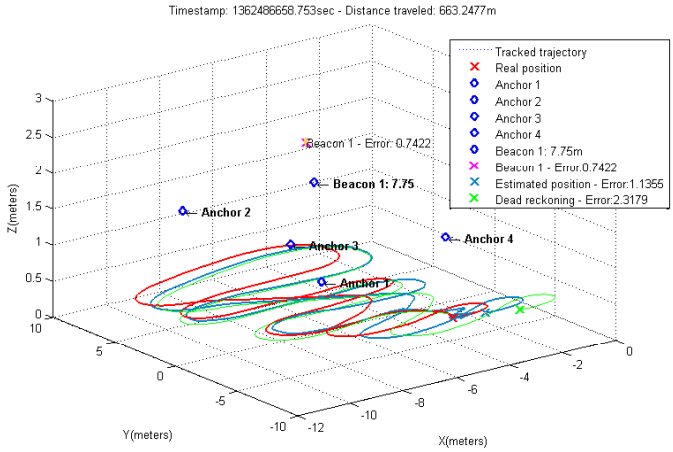


Figure 2.18: 3D representation of the final results with the proposed delayed 3D RO-SLAM. Red line and cross represents ground-truth of the robot. The green line and cross is the dead reckoning estimation. The light blue line and cross represents the localization result of the RO-SLAM algorithm. The real position of the anchors and beacon is represented with a blue diamond, while the estimated position is the pink cross. The number next to the beacon is the distance measurement received from the robot.

$$\begin{aligned} x_r^t &= x_r^{t-1} + v * \delta t * cos(\theta_r^{t-1}) \\ y_r^t &= y_r^{t-1} + v * \delta t * sin(\theta_r^{t-1}) \\ \theta_r^t &= \theta_r^{t-1} + \omega * \delta t \end{aligned} \tag{2.10}$$

where v represents the linear velocity of the vehicle and ω is the steering angle of the vehicle, both measured with optical encoders in a sample time of δt.

For the ground-truth estimation of the robot, an MCL (Monte Carlo Localization) algorithm was used employing a LIDAR sensor (Hokuyo UTM-30LX laser sensor). The real position of beacons was measured with a measuring tape, taking into account the global frame of the CONET testbed.

The results of this experiment using the delayed RO-SLAM algorithm detailed in this chapter are shown in Figure 2.18.

The ground-truth of the robot is represented with a red line and a red cross for the current position, the result of the dead reckoning algorithm is represented

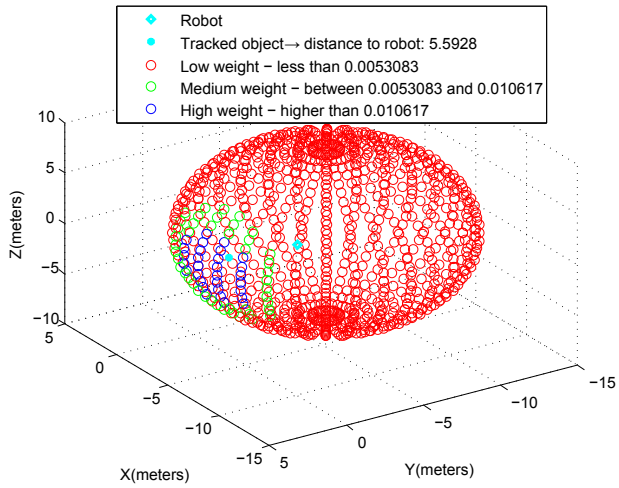


Figure 2.19: Convergence of the PF used to initialize the position of Beacon 1 in the EKF. The most probable particles are depicted with blue circles, the less probable ones are represented with red circles and other intermediate particles are represented with green particles. The cyan point represents the real position of the radio beacon to be mapped and the real position of the robot is represented with a cyan circle.

with the green line and a green cross. The localization result of the RO-SLAM algorithm employed is presented with a light blue line and cross. The real position of the anchors and beacon is represented with a blue diamond, while the estimated position of the beacon is represented with a pink cross, the represented estimation is the estimation of the EKF.

The convergence of the PF is represented in Figure 2.19, in the figure it is depicted the final estimation of the PF from which the EKF estimation is initialized with an initial estimation error of 1.82 meters. On the other hand, the final estimation of the EKF filter has an error of 0.75m, whereas the final estimation error of the robot localization is 1.13 meters.

The difference between the dead reckoning algorithm and the RO-SLAM developed in this section is represented in the 2D representation of the final results in Figure 2.20.

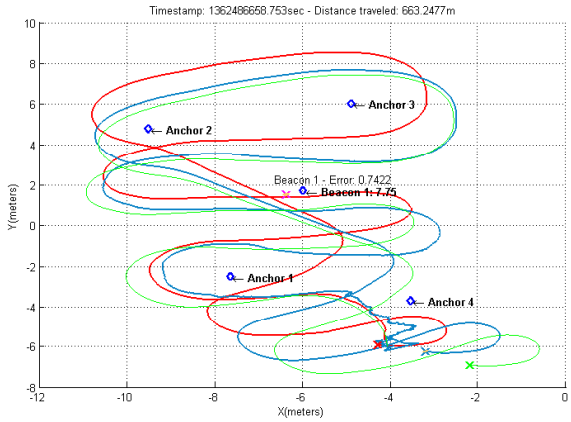


Figure 2.20: 2D representation of the final results with the delayed 3D RO-SLAM developed in this section. The legend of this figure is the same than Figure 2.18.

2.6 Summary and conclusions

This chapter presented an implementation for the RO-SLAM problem applied to 3D environments. The solution presented is based on a delayed initialization of landmarks for 3D RO-SLAM. The initialization strategy is based on a particle filter (PF). The convergence criteria employed in this case is based on the variance of particle set position. Once the landmark is considered to be converged into a single Gaussian, the landmark initial estimation is integrated in a centralized EKF where the localization of the robot takes place. Then, the algorithm is based on a dynamic increment of the EKF state vector as landmarks are initialized. Different simulations have been shown and a real experiment too. As the results show, the main problem of this solution is that the delayed initialization of landmarks does not allow to have a correction of the robot position until they converge, the reason why some static range-only sensors with known position (anchors) have been used to localize the robot until enough landmarks are integrated in the filter. On the other hand, the results show how the landmarks position estimation depends on the accuracy of the particle filter employed for initialization. Wrong landmarks initialization introduces an error on the EKF which in some cases might cause a divergence on the robot position estimation.

Additionally the main drawback of the PF employed to initialize the landmarks

is its known computational complexity when the number of particles is very large and the dependence of the PF accuracy with the number of particles (at least 500 particles are necessary to get a good estimation). This number of particles must be multiplied by the number of beacons to be initialized at the same time in order to get the estimated amount of memory resources required by the algorithm.

On the other hand, this chapter also showed how different measures can be fused in a single filter, making the estimation of the robot position smoother when a GPS is employed.

The chapter also presented the characterization of the range-only sensors employed throughout this dissertation under different situations (indoor/outdoor scenarios, static/dynamic sensor characterization and using ground and aerial robots).

Chapter 3

Efficient undelayed 3D RO-SLAM based on multi-hypotheses

3.1 Introduction

As shown in previous chapter, delayed initialization of landmarks in RO-SLAM approaches are subject to different problems such as not considering inter-beacon measurement until single hypothesis convergence or unknown initialization time of the beacons depending of the robot-sensor trilateration. But, as explained in previous chapter, the main drawback of delayed initialization is that the robot state cannot be corrected unless some complementary sensors are used during initialization stage such as anchors, cameras, etc.

However, since initial beacon probability distribution follows a non-Gaussian distribution, it is difficult to initialize them in a Gaussian filter such as EKF. One possibility consist on using a multi-hypotheses approach by means of Gaussian Mixture Models (also referred as Sum Of Gaussians). But the main drawback of multi-hypotheses frameworks is the computational requirements of these methods.

This chapter extends multi-hypotheses methods by proposing an efficient undelayed scheme for 3D RO-SLAM. The efficiency of the method is mainly related with a reduced parameterization which fits with non-linearities of RO-SLAM and allows to integrate GMM in a EKF-SLAM with a minimal number of parameters, thus reducing the computational complexity of the EKF. Also, some improvements on the correction stage are proposed to avoid the use of known Federated Information Sharing approaches [17, 89].

The chapter also proposes a new robot to landmark and landmark to landmark range-only observation model for EKF which takes advantage of the reduced parameterization. The chapter details how landmark to landmark (inter-landmark) observations can be integrated even when some of the landmarks have not converged to a single hypotheses yet, and how the integration of these observations

reduces the convergence of the filter while at the same time improves the accuracy of the mapping and robot localization.

The method is firstly validated in simulation and later with real experiments using multiple radio-based range-only sensors and an aerial robot.

3.2 Overview and related work

This chapter proposes a general and optimized method for multi-hypotheses solutions which can be either applied to 2D or 3D RO-SLAM. However, the chapter focuses its application to 3D RO-SLAM in order to prove the advantages of using the proposed reduced parameterization with a higher number of multi-modal random variables in the state vector (i.e. not only the hidden azimuth angle variable but also the hidden elevation angle).

In 3D RO-SLAM, given a single range-only observation, the lack of bearing information between sensors, makes the relative position between these elements to follow a uniform spherical shell probability distribution as it is shown in Figure 3.1. Furthermore, in contrast to other schemes like bearing-only SLAM [89], RO-SLAM presents an increased complexity for higher dimensionality (e.g. 3D SLAM in aerial robotics) due to the 1-rank observation model associated to range-only observations (azimuth and elevation angle not observed), against other bearing-only models in which the only unobserved parameter is the distance between the robot and one landmark. Hence, when applying multi-hypotheses approaches, this rank-deficiency implies a higher number of hypotheses/parameters in the state vector.

Thus, in undelayed RO-SLAM approaches, the main research interest resides in how to cope with the initial non-Gaussian distribution of the beacon position as shown in Figure 3.1.

FastSLAM is one of the most common SLAM frameworks which is being applied for efficient undelayed state initialization [8]. However, FastSLAM is based on the Rao-blackwellization principle, which factorize the map assuming no correlation between landmarks. This is also the case of some other decentralized EKF approaches like [28, 100]. However, some applications, like the ARCAS project, include high correlation constraints between landmarks. Thus for example, in the ARCAS project, some structural elements with known geometry include some range-only sensors embedded in them. In this cases, known inter-landmark constraints (i.e. known distances) can be used as an additional

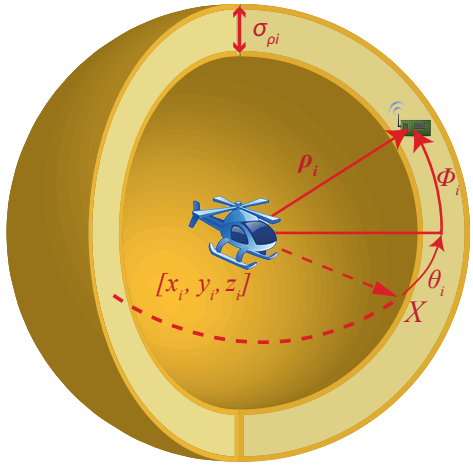


Figure 3.1: Spherical parameterization of a landmark position in 3D RO-SLAM. The yellow area represents the uniform spherical shell distribution where the landmark might be located when receiving a single range-only observation ρ_i between an aerial robot and this landmark. The green object represents the real position of the landmark, whereas the center of the sphere is composed by the position of the aerial robot at the time the range-only observation is received. The thickness of the 3D shell represents the standard deviation of the range measurement σ_{ρ_i}.

observation to fast the convergence of the filter and to get a relative refined accuracy. In this dissertation it is considered the case of a completely centralized state vector. The method proposed here might be applied to a FastSLAM scheme in which the factorization of the map occurs not for single range-only sensors, but for sets of range-only sensors with prior known high correlation constraints. Thus, for example, in ARCAS project, it was possible to keep one centralized EKF for each structural element and without having to keep correlations between those landmarks which are on different structural elements. This will allow to keep the advantages of centralized approaches just on does elements which have this prior constraints, and a decentralized approach for non-constrained landmarks.

The fully centralized approach presented in this dissertation allows to refine the relative landmark's position estimations by integrating inter-landmark observations which will add high cross-covariance terms in the centralized covariance matrix not only between landmarks and robot but also, between pairs of

landmarks.

In order to keep a centralized representation of the SLAM problem, this chapter proposes a reduced parameterization. This reduced parameterization makes possible the application of Gaussian Mixture Models in an EKF-SLAM framework in a more efficient way. The chapter also presents an improvement over the Federated Information Sharing approach used in [17] extended to 3D RO-SLAM, reducing the number of required correction equations of the filter. The chapter also proposes a second correction model based on a single-equation observation model for the multi-hypotheses representation, which avoids to split the information of range-only observations into multiple correction equations as is the case of Federated Information Sharing approaches implemented in [17, 35, 89]. Federated Information Sharing and the two correction schemes detailed in this dissertation are compared with simulated an real experiments to show their feasibility.

For the integration of inter-landmark observations, some authors have already included them in their RO-SLAM formulation [27, 30, 94, 96]. These range-only observations not only fast the convergence of the multi-modal distribution of landmarks but also improve the mapping accuracy. This redundant information of landmarks increases both reliability and precision of the map as demonstrated in [19] for multi-sensor fusion in SLAM and, indirectly, the localization of the robot through its correlation with landmarks. For this reason, range-only observations between landmarks might become specially informative in a centralized framework [17] like EKF-SLAM. However, inter-landmark observations are usually integrated in the filter once both landmark position estimations converge to a single solution, without taking advantage of this information before. In [98], inter-landmark observations are used in a efficient way to fast the convergence of the landmarks initialization stage but under a delayed initialization approach. This chapter explains a new observation model for inter-landmark range-only measurements which allows to integrate these measurements since the very beginning in an efficient and undelayed way, fasting the convergence of the landmarks without loosing information from initial observations. The chapter details a correction scheme for the integration of inter-landmark observations even when the hypotheses have not converged to a single solution yet.

The rest of the chapter will first introduce the reduced landmarks parameterizations and will compare with the different landmark parameterizations proposed in the literature. Later, the chapter details each of the stages of the EKF-SLAM

approach proposed.

3.3 Reduced parameterization in a nutshell

This section makes a comparison between the reduced parameterization proposed in this dissertation with other RO-SLAM parameterizations used in the literature for multi-hypotheses approaches. Two important aspects should be taken into account when comparing the parameterizations used for an EKF-SLAM approach: first, the computational complexity of the Gaussian filters, which, in the case of EKF, is highly dependent on the number of parameters of the state vector [92], and second, the scalability of the system with the number of landmarks.

The most common parameterization used in the literature for landmarks position is the Cartesian parameterization [8, 80]. In this parameterization, each hypotheses j of a landmark i is composed by Cartesian coordinates so that the total number of parameters for H hypotheses would be $2H$ ($f_{ij} = [x_{ij}, y_{ij}]^T$) for 2D and $3H$ ($f_{ij} = [x_{ij}, y_{ij}, z_{ij}]^T$) for 3D.

On the other hand, [26] proposed a polar parameterization for 2D RO-SLAM where each landmark position is parametrized as $f_i = [x_i, y_i, \rho_i, \theta_i]$. Being, x_i and y_i the center of the initial annulus distribution of the landmark position from which the first range-only observation is received with a radius of ρ_i meters and the angle θ_i between the reference frame of the robot and landmark i. This polar parameterization fits better with 2D RO-SLAM since it allows to represent the annular distribution of a single range-only observation using an elongated Gaussian in polar coordinates ($\rho\theta$-space). However, to represent the flip ambiguity which appears with the second range-only observation, the authors use a heuristic method to split the unimodal distribution into two Gaussian distributions as a result of the intersection between the first annulus distribution and the second generated with a new range-only observation. Thus, to represent these 2 hypotheses, polar parameterization uses 8 parameters (4 for each hypotheses) against 4 parameters needed in Cartesian parameterization for the same number of hypotheses. One of the main drawbacks of this parameterization is that it duplicates the common parameters x_i, y_i and ρ_i in both hypotheses. On the other hand, it requires to delay the initialization of both hypotheses until a good trilateration is achieved to split the initial Gaussian distribution into two Gaussians. This trilateration of the hypotheses is specially difficult in the case of 3D RO-SLAM where ambiguities are made worse. When inter-node range-only observations are integrated [27]

this polar parameterization requires up to 5×2^m parameters to represent the complete multi-modal distribution with m being the number of landmarks. In addition, as [27] is based on a decentralized solution, it does not take into account the correlations between landmarks.

An extension of this polar parameterization was proposed in [17] for 2D RO-SLAM, in this case the authors use a Gaussian Mixture Model (GMM) to model the annular distribution of the landmark position belief. GMMs are probability distributions that are convex combination of Gaussian distributions, they form a semi-parametric alternative to non-parametric distributions, providing a better flexibility and precision when modeling the underlying statistics of range-only observations. In a GMM each mode i is a normal distribution $\mathcal{N}(\mu_i, \sigma_i)$ weighted by ω_i, where $0 \leq \omega_i \leq 1$ and $\sum_{i=1}^{k} \omega_i = 1$. Then, the Gaussian Mixture probability distribution for N modes, looks like:

$$f_X(x) = \sum_{j=1}^{N} \omega_j \mathcal{N}(x; \mu_j, \sigma_j) \tag{3.1}$$

With this parameterization, each landmark state is parametrized as $f_i = [x_i, y_i, \rho_i, \theta_{i1}, ..., \theta_{iN}]^T$. Thus, when the number of hypotheses is $H \geq 4$, the number of parameters of this reduced polar parameterization is shorter with respect classical Cartesian one ($3 + H$ against $2H$ used in Cartesian parameterization). The advantage of this reduced polar parameterization with respect the polar presented in [26] is that it does not duplicate common hypotheses parameters x_i, y_i and ρ_i. Furthermore, unlike decentralized filters [27], the single state-vector-parameterization of nodes used in [17] allows to take into account the correlations between landmarks.

However, as in RO-SLAM the number of hidden variables to be estimated increases with the dimensionality of the problem, for the case of 3D RO-SLAM, it is required to increase the state vector of landmarks to estimate not only the azimuth angle of the landmark but also the elevation angle (see Figure 3.1). A straightforward extension of previous polar parameterizations into spherical parameterization would consist on using a single GMM with multivariate Gaussian modes. Each mode would represent a single hypothesis with azimuth angle θ_i and elevation angle ϕ_i. Then, the 3D state vector of a range-only landmark with Cartesian (3.2) and the spherical (3.3) parameterization described would be:

$$f_i = [x_{i1}, y_{i1}, z_{i1}, x_{i2}, y_{i2}, z_{i2}, ..., x_{iH}, y_{iH}, z_{iH}]^T \quad (3.2)$$

$$f_i = [x_i, y_i, z_i, \rho_i, \theta_{i1}, \phi_{i1}, \theta_{i2}, \phi_{i2}, ..., \theta_{iH}, \phi_{iH}]^T \quad (3.3)$$

Then, the required number of parameters per landmark with a 3D Cartesian parameterization (3.2) would be $3H$, whereas with this single-GMM spherical formulation (3.3) would be $4 + 2H$, being H the number of hypotheses.

The reduced parameterization proposed here is based on a generalization of the spherical parameterization (3.3) which makes it suitable for different dimensionalities. This reduced parameterization, instead of representing all hypotheses with a single GMM, it uses one GMM per hidden variable (i.e. in RO-SLAM, one GMM for each bearing parameter). Thus in the case of 3D RO-SLAM, hypotheses used to cover the spherical shell distribution shown in Figure 3.2 are parametrized using 3 parameters for the center $[x_i, y_i, z_i]^T$, another for the radius of the sphere ρ_i, N parameters to represent the azimuth angle samples (modes of the first GMM θ) and M parameters for elevation angle samples (modes of the GMM used for ϕ angle). Thus, the number of parameters used for a landmark in 3D RO-SLAM would be $4 + N + M$, being the total number of hypotheses $H = N \times M$. In the case of 2D RO-SLAM, would just use one GMM for azimuth angle as in [17] with $4 + N$ parameters.

Then, for 3D parameterization, the complete state vector of a landmark i using this reduced parameterization would be:

$$f_i = [x_i, y_i, z_i, \rho_i, \theta_{i1}, ..., \theta_{iN}, \phi_{i1}, ..., \phi_{iM}]^T \quad (3.4)$$

Table 3.1 and Figure 3.3 shows a comparison in the number of parameters required to represent the spherical shell distribution shown in Figure 3.1 with the 3D Cartesian, the single-GMM spherical and the reduced parameterization for different number of hypotheses. As can be seen, for example, to represent **1024 hypotheses** it is required 3072 parameters in the case of 3D Cartesian, 2052 in the case of the single-GMM spherical and only **68 parameters** with the proposed reduced parameterization (4 plus 32 azimuth + 32 elevation angles - Section 3.4 shows the details of state initialization).

However, the reduction of parameters in the reduced spherical parameterization limits the distribution of the hypotheses, i.e. while the azimuth and elevation samples are distributed uniformly on the range $0 - 2\pi$, the joint hypotheses distribution is not uniformly distributed. Then, a good covariance should be used for

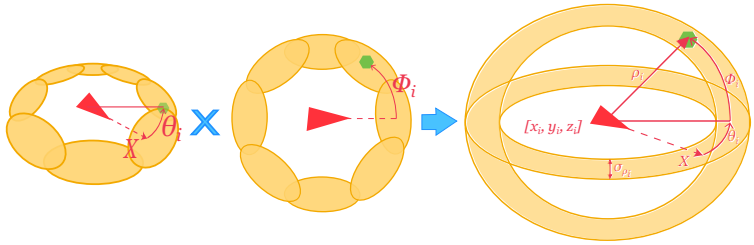


Figure 3.2: Reduced parameterization for 3D RO-SLAM: The combination of the GMM used for the azimuth angle with the GMM used for the elevation angle represent the set of hypotheses used to model the complete spherical shell distribution.

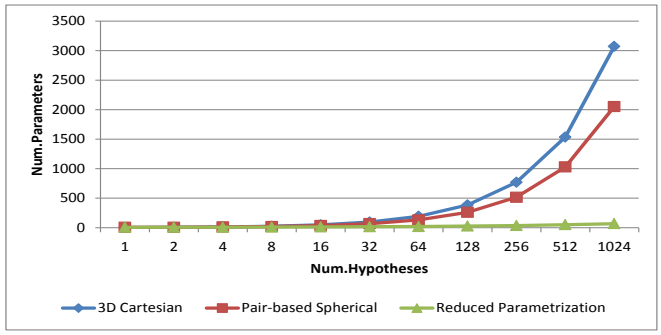


Figure 3.3: Number of parameters used to model a given quantity of hypotheses for the different 3D RO-SLAM parameterizations.

the azimuth and elevation angles for each Gaussian to cover the entire spherical shell distribution of the landmark position. The way in which these covariance matrices should be computed during the initialization stage of a landmark will be shown in the following section.

3.4 3D Range-only EKF-SLAM

As in [17], this approach uses an EKF-SLAM framework to solve the 3D RO-SLAM problem. EKF-SLAM gathers in a single state vector the robot and landmarks parameters, using the covariance matrix to represent the correlations between robot and landmarks. The state vector of the proposed EKF-SLAM for 3D RO-SLAM is composed by the following parameters:

Table 3.1: Comparison of different parameterizations for 3D RO-SLAM.

#Hyp\#Params	3D Cart.	Spherical	Reduced
1	3	6	6
2	6	8	7
4	12	12	8
8	24	20	10
16	48	36	12
32	96	68	16
64	192	132	20
128	384	260	28
256	768	516	36
512	1536	1028	50
1024	3072	2052	68

$$\mathbf{x}^t = [\mathbf{x}_r^t, \mathbf{f}_1^t, \mathbf{f}_2^t, \ldots, \mathbf{f}_m^t]^T \tag{3.5}$$

where $\mathbf{x}_r^t$ is the robot state which follows an unimodal Gaussian distribution (e.g. $\mathbf{x}_r^t = [x_r^t, y_r^t, z_r^t]^T$) and $\mathbf{f}_i^t$ is the landmark i multi-modal Gaussian state.

For the prediction stage, it can be used any Gaussian dynamic model according to the mobile robot employed for RO-SLAM. However, as this thesis is focused on aerial vehicles, in this case it is proposed the dynamic model for quad-copter described in [11].

On the other hand, for the update stage the whole algorithm is summarized in Figure 3.4. In the first step of this flow chart the reader may notice that range-only observations are pre-filtered before passing them to the EKF-SLAM framework. This pre-filtering is highly recommended to filter range-only outliers which might lead to EKF divergences. The pre-filtering algorithm used in this dissertation is detailed in Chapter 4. The other steps of the flow chart are detailed in the following subsections.

3.4.1 Undelayed initialization

For initialization of new landmarks it is proposed an adaptive scheme which adapts to the first range-only observation received at current robot position. Thus,

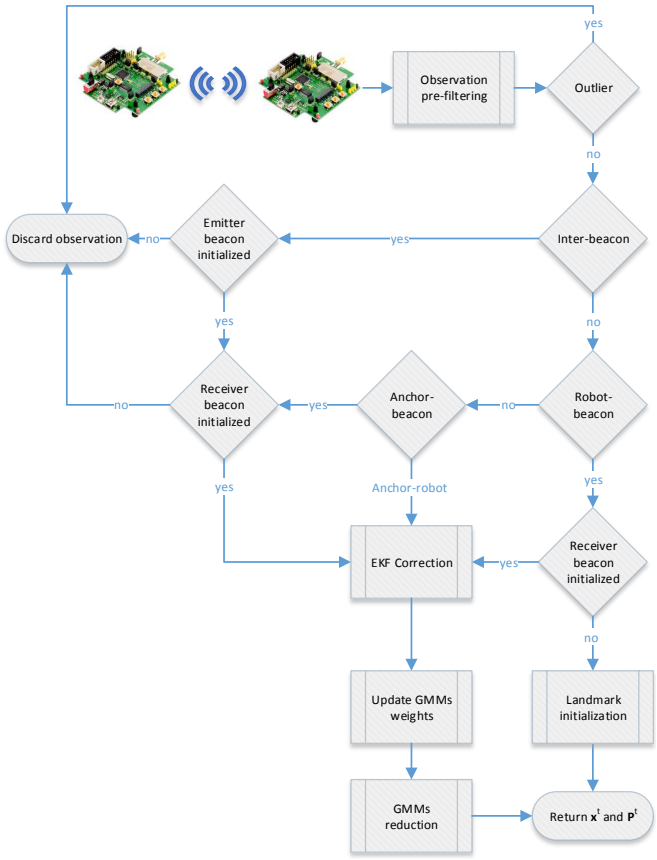


Figure 3.4: Flow chart describing the update stage of the proposed RO-SLAM algorithm upon receiving range measurements.

only range measurements between robot and one landmark are considered for initialization as shown in Figure 3.4. Landmarks located further than a certain distance threshold should be discarded to reduce the computational load (remember that in some cases 2.9 range information worsen with distance).

New landmarks are initialized when the first range-only observation r_i^t is received by the robot from a landmark i. With this observation r_i^t and the current position of the robot $\mathbf{x}_r^t$, the parameters of (3.4) are initialized as:

$$\mathbf{x}_i^t = \mathbf{x}_r^t \tag{3.6}$$
$$\rho_i^t = r_i^t \tag{3.7}$$
$$\theta_{ij}^t = \frac{2\pi j}{N} - \pi \quad j = 1, \ldots, N \tag{3.8}$$
$$\phi_{ij}^t = \frac{\pi j}{M} - \frac{\pi(M+1)}{2M} \quad j = 1, \ldots, M \tag{3.9}$$

Being $\mathbf{x}_i^t = [x_i, y_i, z_i]$ the robot position at the time t[1] the range-only observation r_i^t is received.

With (3.8) and (3.9), N azimuth and M elevation Gaussian modes are uniformly distributed within ranges $(\text{-}\pi, \pi]$ and $(\text{-}\pi/2, \pi/2)$ respectively. However, the number of modes required to distribute all hypotheses in the spherical shell distribution depends on the radius of the sphere ρ_i and a desired density of hypotheses d (in practice $d = 0.18$ gives good results). Thus, the appropriate number of azimuth and elevation modes for both GMMs to cover the spherical shell distribution shown in Figure 3.1 might be computed from

$$H^* = 4\pi r_i^2 d \tag{3.10}$$

Given H^*, it is possible to extract the required number of azimuth N and elevation M samples from constraints (3.10) and $H^* = N \times M$. Then, as hypotheses should be distributed to cover a spherical shell distribution, the number of elevation samples required are $M = N/2$ and hence the number of azimuth samples can be computed from the last two expressions as $N = \lceil \sqrt{2H^*} \rceil$. With this initialization strategy, the actual number of hypotheses generated is $H = N \times M \geq H^*$.

The next step is to initialize the covariance matrix of each Gaussian mode and their associated weights, $\omega_{\theta_{in}}$ and $\omega_{\phi_{im}}$. As both GMMs should approximate an uniform distribution around the azimuth and elevation space, the values of $\omega_{\theta_{in}}$ and $\omega_{\phi_{im}}$ are easily initialized as $\omega_{\theta_{in}} = 1/N$ and $\omega_{\phi_{im}} = 1/M$.

The standard deviation of each variable of the state vector $\mathbf{f}_i$ is initialized as follows. The covariance matrix of parameters x_i, y_i and z_i is initialized using the current covariance of the robot position, the variance of ρ_i is initialized using the standard deviation of the range measurement as $\sigma_{\rho_i}^2 = \sigma_{r_i}^2$ (correlations with

[1] Sometimes t is omitted when not relevant to simplify equations.

this variable are set to 0), finally, the standard deviation of each Gaussian mode θ_{in} and ϕ_{im} is identically initialized for each GMM according to the following expressions (setting initial correlations to 0):

$$\sigma_{\theta_{in}} = \frac{2\pi}{k_\theta N} \quad n = 1, \ldots, N \tag{3.11}$$

$$\sigma_{\phi_{im}} = \frac{\pi}{k_\phi M} \quad m = 1, \ldots, M \tag{3.12}$$

where values k_θ and k_ϕ of the expressions (3.11) and (3.12) are proportional factors computed using the Kullback-Leibler (KL) distance between a Gaussian Mixtures and a target uniform distributions for θ_i and ϕ_i multi-modal variables. The KL divergence factor is a statistic which comes from information theory and measures the amount of additional information that is required to model a target distribution (in this case the uniform distribution) given a proposal distribution (in this case a Gaussian Mixture). However, this statistic does not have a closed form for Gaussian Mixtures, hence a Monte Carlo sampling method [57] can be used to get an approximation of this statistic. When comparing two probability distributions, the best fit is that which has a information divergence (KL distance) equal to 0. Then, k_θ and k_ϕ were computed from a set of simulations where different Gaussian Mixtures with different number of modes were used to model a uniform distribution over $(\text{-}\pi, \pi]$ for k_θ and $(\text{-}\pi/2, \pi/2)$ for k_ϕ. For each Gaussian Mixture with k modes, different standard deviations were tested to initiate each mode of the Gaussian Mixture. The standard deviations which had the closest KL distance to 0 were selected as the optimal deviation for a given Gaussian Mixture with k modes. This values are marked with circles in Figure 3.5a and Figure 3.5c. Then, the selected standard deviations were used to get the mean value of k_θ and k_ϕ using the expressions (3.11) and (3.12). The results of these experiments are depicted in Figure 3.5a and Figure 3.5c, where the X axis represents the different standard deviations used for each Gaussian Mixture with k modes and the Y axis represents the KL distance. The values that gave the best fit were $k_\theta = 1.7$ and $k_\phi = 2.5$. Figure 3.5b and Figure 3.5d shows an example of the GMM generated for a range measurement of $r = 5m$ received at position $[0, 0, 0]$ using the initialization strategy explained in this section and proportional factors k_θ and k_ϕ. The combination of both GMMs, Figure 3.5b and Figure 3.5d, results in a uniform spherical shell distribution.

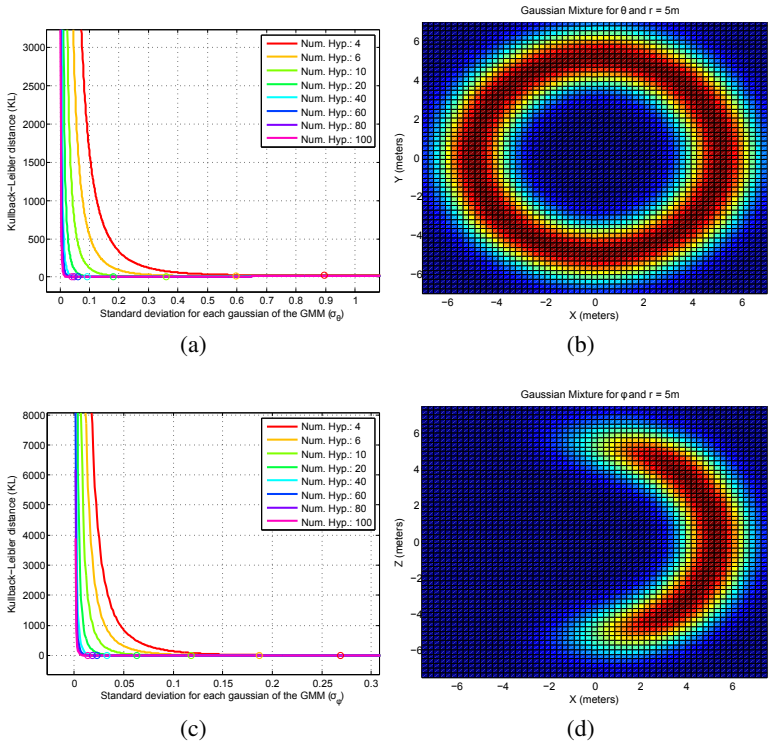


Figure 3.5: Experiments performed for the optimal selection of k_θ and k_ϕ factors. In (a) and (c) the Y axis represents the Kullback-Leibler divergence factor and the X axis represents the standard deviation used for each simulated GMM. Each series represents the simulation for a fixed number of modes in the GMM. (b) and (d) shows an example of the GMM generated using the optimal values of K_θ and K_ϕ respectively. The combination of both GMM, (b) and (d), gives the uniform spherical shell distribution for a range measurement $r = 5m$ received at coordinates $[0, 0, 0]$.

Once a landmark is initialized, its state and covariance are appended at the end of the current state vector $\mathbf{x}^t$ and covariance matrix $\mathbf{P}^t$ of the EKF-SLAM. The correlations of the whole covariance matrix $\mathbf{P}^t$ between the new landmark covariance matrix $\mathbf{P}_{f_i}$ and the robot covariance $\mathbf{P}_r$, and also between the new landmark and the old ones, are computed as indicated in the following expression:

$$P = \begin{pmatrix} \mathbf{P}_r & \mathbf{P}_{rf_1} & \cdots & \mathbf{P}_{rf_{i\text{-}1}} & \mathbf{P}_r\mathbf{J}_\oplus^T \\ \mathbf{P}_{rf_1}^T & \mathbf{P}_{f_1} & \cdots & \mathbf{P}_{f_1f_{i\text{-}1}} & \mathbf{P}_{rf_1}^T\mathbf{J}_\oplus^T \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{P}_{rf_{i\text{-}1}}^T & \mathbf{P}_{f_1f_{i\text{-}1}}^T & \cdots & \mathbf{P}_{f_{i\text{-}1}} & \mathbf{P}_{rf_{i\text{-}1}}^T\mathbf{J}_\oplus^T \\ \mathbf{J}_\oplus\mathbf{P}_r & \mathbf{J}_\oplus\mathbf{P}_{rf_1} & \cdots & \mathbf{J}_\oplus\mathbf{P}_{rf_{i\text{-}1}} & \mathbf{P}_{f_i} \end{pmatrix}$$

being $\mathbf{J}_\oplus$ the Jacobian of the equations which relates the robot position with landmarks (3.6), (3.7), (3.8) and (3.9). As the equations which relate a landmark with the position of the robot are a simple assignation, the Jacobian $\mathbf{J}_\oplus$ is a Fx3 matrix of zeros except the initial 3x3 submatrix which corresponds to the identity matrix, where F is the size of the new landmark state vector.

$$\mathbf{J}_\oplus = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ \vdots & \vdots & \vdots \\ 0 & 0 & 0 \end{bmatrix}.$$

3.4.2 Correction stage: robot-landmark observations

This section focuses in the correction stage of the EKF-SLAM framework for robot to landmark i range-only observations (denoted as r_i).

As range-only observations between robot and anchors are independent from the map estimations, this section will first introduce the correction stage for this kind of measurements.

For landmarks' observations it is considered that at least one range-only observation for landmark i has already been received to initialize this landmark on the state vector and hence the equations detailed here are used for the correction of the filter and to update the weights of the landmark.

Robot to anchors corrections

The observation model for robot-anchor range-only measurements r_{A_i} is independent from the map parameterization since it only affects the robot state vector. This particular observation model is the same used for EKF localization methods (i.e. no SLAM). The observation function $h(\mathbf{x}^t)$ for this robot-anchor observations is as follows:

$$h(\mathbf{x}^t) = \sqrt{(x_r^t - x_{A_i})^2 + (y_r^t - y_{A_i})^2 + (z_r^t - z_{A_i})^2} \tag{3.13}$$

where $\mathbf{x_{A_i}} = \{x_{A_i}, y_{A_i}, z_{A_i}\}$ represents the static position of anchor i[2].

The Jacobian for this observation model is:

$$\mathbf{H} = \left[\frac{(x_r^t - x_{A_i})}{\mathbf{x}^t}, \frac{(y_r^t - y_{A_i})}{h(\mathbf{x}^t)}, \frac{(z_r^t - z_{A_i})}{h(\mathbf{x}^t)}\right] \tag{3.14}$$

The variance for this observation model is equal to the variance of the range-only measurement r_{rA_i}, i.e. $\sigma^2_{r_{A_i}}$.

Correction method 1: Federated Information Sharing

As already introduced, Federated Information Sharing [89] is one of the most implemented solutions for multi-hypotheses approaches using EKF-SLAM framework. This method applies the Principle of Measurement Reproduction [99] to overcome filter inconsistencies due to the integration of redundant information. This principle is used when a single source of information (i.e. a single range-only observation) must be divided as if it was generated from multiple sources of information (i.e. each of the landmark hypotheses). The principle is based on the fact that the correction of the estimate of a random variable by a set of measurement tuples $\{\mathbf{y}; \mathbf{R}_j\}$ is equivalent to the unique correction by $\{\mathbf{y}; \mathbf{R}\}$ if

$$\mathbf{R^{-1}} = \sum \mathbf{R}_j^{-1} \tag{3.15}$$

The first method proposed in this dissertation is based on Federated Information Sharing approach used in [17, 89] but with a reduction on the number of equations (i.e. hypothetical sources of information). The first method proposed here uses one correction equation per angle sample θ_{in} or ϕ_{im} (i.e. $N + M$

[2]The temporal index t is not included in anchor parameters since they not change over time. Indeed they are not part of the vector state.

equations) instead of one per hypothesis $\{\theta_{in}; \phi_{im}\}$ (i.e. $N \times M$ equations)as is the case of classical Federated Information Sharing. This first method is referred here as **Multi-Hypotheses Correction (MHC)**, whereas the classical Federated Information Sharing approach using full hypothesis $\{\theta_{in}; \phi_{im}\}$ is referred as **Full Hypothesis Correction (FHC)**.

Thus, reducing the number of equations in the proposed correction scheme, the computational load required to compute the EKF innovation (or residual) matrix is also reduced. Notice, that the main computational complexity of EKF depends on two aspects: first, the size of the state vector, which here has been reduced with the proposed Reduced Parameterization, and second, on the computation of the inverse innovation matrix, here reduced from $N \times M$ to $N + M$ equations. Each of these $N + M$ equations is of the form:

$$h_j(\mathbf{x}) = \sqrt{\delta_x^2 + \delta_y^2 + \delta_z^2} \tag{3.16}$$

where $\delta_x = (x_{f_{ij}} - x_r)$, $\delta_y = (y_{f_{ij}} - y_r)$, $\delta_z = (z_{f_{ij}} - z_r)$. For the first N equations, variables $x_{f_{ij}}$, $y_{f_{ij}}$ and $z_{f_{ij}}$ are computed as:

$$\begin{aligned} x_{f_{ij}} &= x_i + \rho_i cos(\theta_{in}) cos(\phi_i) \\ y_{f_{ij}} &= y_i + \rho_i sin(\theta_{in}) cos(\phi_i) \\ z_{f_{ij}} &= z_i + \rho_i sin(\phi_i) \end{aligned} \tag{3.17}$$

for $j = 1 \dots N$ and with $n = 1 \dots N$. ϕ_i is the simplified notation to represent the expected value of elevation angle Gaussian Mixture $E[\phi_i]$. The general expression used to calculate the expectation of a GMM is:

$$E[GMM] = \sum_{i=1}^{k} \omega_i \mu_i \tag{3.18}$$

Similarly, the rest of M equations are calculated with the following expressions for variables $x_{f_{ij}}$, $y_{f_{ij}}$ and $z_{f_{ij}}$ as:

$$\begin{aligned} x_{f_{i,j}} &= x_i + \rho_i cos(\theta_i) cos(\phi_{im}) \\ y_{f_{i,j}} &= y_i + \rho_i sin(\theta_i) cos(\phi_{im}) \\ z_{f_{i,j}} &= z_i + \rho_i sin(\phi_{im}) \end{aligned} \tag{3.19}$$

for $j = N + 1 \dots N + M$ and with $m = 1 \dots M$ and $\theta_i = E[\theta_i]$ the expectation of the Gaussian Mixture used to model the azimuth parameter.

For Jacobian H_j of each equation $h_j(\mathbf{x})$, the expression would be:

$$H_j = \left[\frac{\partial h(\mathbf{x})}{\partial \mathbf{x}_r}, 0, \ldots, 0, H_{ij}, 0, \ldots, 0\right] \tag{3.20}$$

where the Jacobian associated to the robot state would be:

$$\frac{\partial h(\mathbf{x})}{\partial \mathbf{x}_r} = \left[\frac{-\delta_x}{h(\mathbf{x})}, \frac{-\delta_y}{h(\mathbf{x})}, \frac{-\delta_z}{h(\mathbf{x})}\right] \tag{3.21}$$

the Jacobian H_{ij} associated to landmark i and azimuth value θ_{in} would be:

$$\begin{aligned} H_{ij} = [&\frac{\delta_x}{h(\mathbf{x})}, \frac{\delta_y}{h(\mathbf{x})}, \frac{\delta_z}{h(\mathbf{x})}, \\ &\frac{c(\theta_{in})c(\phi_i)\delta_x + s(\theta_{in})c(\phi_i)\delta_y + s(\phi_i)\delta_z}{h(\mathbf{x})}, \\ &0, \ldots, \delta\theta_{in}, \ldots, 0, \\ &\omega_{\phi_{i1}}\delta\phi_i, \ldots, \omega_{\phi_{im}}\delta\phi_i] \end{aligned} \tag{3.22}$$

the Jacobian H_{ij} associated to landmark i and elevation value ϕ_{im} would be:

$$\begin{aligned} H_{ij} = [&\frac{\delta_x}{h(\mathbf{x})}, \frac{\delta_y}{h(\mathbf{x})}, \frac{\delta_z}{h(\mathbf{x})}, \\ &\frac{c(\theta_i)c(\phi_{im})\delta_x + s(\theta_i)c(\phi_{im})\delta_y + s(\phi_{im})\delta_z}{h(\mathbf{x})}, \\ &\omega_{\theta_{i1}}\delta\theta_i, \ldots, \omega_{\theta_{in}}\delta\theta_i, \\ &0, \ldots, \delta\phi_{im}, \ldots, 0] \end{aligned} \tag{3.23}$$

being

$$\delta\theta_i = -\frac{\rho_i s(\theta_i)c(\phi_i)\delta_x - \rho_i c(\theta_i)c(\phi_i)\delta_y}{h(\mathbf{x})} \tag{3.24}$$

$$\delta\phi_i = -\frac{\rho_i c(\theta_i)s(\phi_i)\delta_x + \rho_i s(\theta_i)s(\phi_i)\delta_y - \rho_i c(\phi_i)\delta_z}{h(\mathbf{x})} \tag{3.25}$$

$$\delta\theta_{in} = -\frac{\rho_i s(\theta_{in})c(\phi_i)\delta_x - \rho_i c(\theta_{in})c(\phi_i)\delta_y}{h(\mathbf{x})} \tag{3.26}$$

$$\delta\phi_{im} = -\frac{\rho_i c(\theta_i)s(\phi_{im})\delta_x + \rho_i s(\theta_i)s(\phi_{im})\delta_y - \rho_i c(\phi_{im})\delta_z}{h(\mathbf{x})} \tag{3.27}$$

where $c(x) = cos(x)$ and $s(x) = sin(x)$.

The last matrix to be computed is the noise of the observation model:

$$Q = \begin{bmatrix} \sigma^2_{r_i 1} & 0 & \dots & 0 \\ 0 & \ddots & \dots & 0 \\ 0 & \dots & \sigma^2_{r_i N+M-1} & 0 \\ 0 & \dots & 0 & \sigma^2_{r_i N+M} \end{bmatrix} \tag{3.28}$$

The terms of this noise matrix, as introduced before, are calculated using the Principle of Measurement Reproduction. So the range-only measurement variance $\sigma^2_{r_i}$ is splitted following (3.15) as:

$$(\sigma^2_{r_i})^{-1} = \sum_{j=1}^{N+M} (\sigma^2_{r_i j})^{-1} \tag{3.29}$$

where each individual variance is calculated as $\sigma^2_{r_i j} = \sigma^2_{r_i}/\lambda_j$ being:

$$\lambda_j = \begin{cases} p(r_i|\mathbf{x}^t_r, \mathbf{x}^t_i, \rho_i, \theta_{in}, \phi_i) & j \in [1, N] \quad \& \quad n = j \\ p(r_i|\mathbf{x}^t_r, \mathbf{x}^t_i, \rho_i, \theta_i, \phi_{im}) & j \in [N+1, N+M] \quad \& \quad m = j - N \end{cases} \tag{3.30}$$

so that all these values are normalized as $\sum \lambda_j = 1$.

Correction method 2: Gaussian Mixture Correction (GMC)

Later, the first correction scheme was improved without much loss of accuracy to a single correction equation using a novel technique referred as Gaussian Mixture Correction (GMC). By using this technique, the inversion of the innovation matrix becomes a scalar inversion when receiving a single range measurement at time t. The technique consists on the integration of the complete GMMs in the correction equation by using the expectation of both Gaussian Mixtures, so that the weights of the GMMs are taken into account as constant variables when correcting the landmarks hypotheses state. Thus, using GMC in 3D RO-SLAM is as easy as using the expectation (3.18) of the azimuth $E[\theta_i] = \theta_i$ and elevation $E[\phi_i] = \phi_i$ GMMs in the non-linear observation model of range measurements. Then, the integration of expectations θ_i, ϕ_i in the range-only observation model for the linearization point $\mathbf{x} = [x_r, y_r, z_r, x_i, y_i, z_i, \rho_i, \theta_i, \phi_i]$ can be expressed as:

$$h(\mathbf{x}) = \sqrt{\delta_x^2 + \delta_y^2 + \delta_z^2} \tag{3.31}$$

where $\delta_x = (x_{f_i} - x_r)$, $\delta_y = (y_{f_i} - y_r)$, $\delta_z = (z_{f_i} - z_r)$ and x_{f_i}, y_{f_i} and z_{f_i} stand for:

$$\begin{aligned} x_{f_i} = &\quad x_i + \rho_i cos(\theta_i) cos(\phi_i) \\ y_{f_i} = &\quad y_i + \rho_i sin(\theta_i) cos(\phi_i) \\ z_{f_i} = &\quad\quad z_i + \rho_i sin(\phi_i) \end{aligned} \tag{3.32}$$

Notice that, in contrast to the MHC method, the use of this observation model for 2D RO-SLAM is similar to 3D RO-SLAM but omitting altitude terms z and with a fixed elevation value $\phi_i = 0$ in (3.31) and (3.32).

For Jacobian H of $h(\mathbf{x})$, the expression would be:

$$H = \left[\frac{\partial h(\mathbf{x})}{\partial \mathbf{x}_r}, 0, \ldots, 0, H_i, 0, \ldots, 0 \right] \tag{3.33}$$

where the Jacobian associated to the robot state would be:

$$\frac{\partial h(\mathbf{x})}{\partial \mathbf{x}_r} = \left[\frac{-\delta_x}{h(\mathbf{x})}, \frac{-\delta_y}{h(\mathbf{x})}, \frac{-\delta_z}{h(\mathbf{x})} \right] \tag{3.34}$$

and the Jacobian of landmark i is:

$$\begin{aligned} H_i = [& \frac{\delta_x}{h(\mathbf{x})}, \frac{\delta_y}{h(\mathbf{x})}, \frac{\delta_z}{h(\mathbf{x})}, \\ & \frac{c(\theta_i)c(\phi_i)\delta_x + s(\theta_i)c(\phi_i)\delta_y + s(\phi_i)\delta_z}{h(\mathbf{x})}, \\ & \omega_{\theta_{i1}}\delta\theta_i, \ldots, \omega_{\theta_{in}}\delta\theta_i, \\ & \omega_{\phi_{i1}}\delta\phi_i, \ldots, \omega_{\phi_{im}}\delta\phi_i] \end{aligned} \tag{3.35}$$

being

$$\delta\theta_i = \quad -\frac{\rho_i s(\theta_i)c(\phi_i)\delta_x - \rho_i c(\theta_i)c(\phi_i)\delta_y}{h(\mathbf{x})} \tag{3.36}$$

$$\delta\phi_i = \quad -\frac{\rho_i c(\theta_i)s(\phi_i)\delta_x + \rho_i s(\theta_i)s(\phi_i)\delta_y - \rho_i c(\phi_i)\delta_z}{h(\mathbf{x})} \tag{3.37}$$

where $c(x) = cos(x)$ and $s(x) = sin(x)$.

With this correction scheme, it is no longer necessary to split the variance of the range measurement among multiple equations using Federated Information Sharing approach. In general, including the weights of GMMs in the observation model $h(\mathbf{x})$ makes it more informative than MHC method, allowing a faster convergence in the filter. With GMC approach all hypotheses are equally affected when they are uniformly distributed (i.e. all hypotheses have the same weight) but, as the weights are updated, those hypotheses with higher likelihood are favored, making the whole GMMs converge to the most probable hypotheses. GMC not only reduces the computational load required in the correction stage of multi-modal observation models but also eases the implementation of multi-hypotheses solutions with respect other similar methods in the literature.

However, the efficiency and simplicity of this correction schemes comes at expense of possible initial loss of accuracy on the correction scheme with respect other approaches. This little loss of accuracy only happens in those cases where the initial GMM distributions makes the Jacobian (3.31) linearization point to be far from the actual position of the landmark since this linearization point is computed as the expectation value of each GMM. However, as will be shown later, this correction scheme compensates this loss of accuracy by accelerating the convergence time of hypotheses with respect other approaches and hence making the linearization point converge to the real landmark position with only a few range-only observations.

Weights update method 1: Maximum Likelihood Update (MLU)

After the EKF state has been corrected, the next step consist on updating the weights of the GMMs. Classical approaches in the literature have a unique weight ω_{inm} per full hypothesis $\{\theta_{in}; \phi_{im}\}$. Hence, these weights are updated using the following expression:

$$\omega_{inm} = p(r_i^t | \mathbf{x}_r^t, \mathbf{x}_i^t, \rho_i^t, \theta_{in}^t, \phi_{im}^t) \tag{3.38}$$

This classical approach is referred here as **Full Hypothesis Update (FHU)**. FHU might be applied to the reduced spherical parameterization by just keeping the weights of each full hypothesis. If Reduced Parametrization is used with FHU, then to recover the marginal probability of $\theta_i n$ or $\phi_i m$ GM modes it is necessary to make use of Total Probability Theorem:

$$\omega_{\theta_{in}} = \sum_{j=1}^{M} \omega_{inj} \tag{3.39}$$

$$\omega_{\phi_{im}} = \sum_{j=1}^{N} \omega_{ijm} \tag{3.40}$$

Additionally, as neither, the reduced parameterization nor the new GMC observation model, imply an independence between azimuth and elevation angles, the parameterization might also be used with classical update strategies used in the literature by storing the joint hypotheses weights instead of storing the weights of azimuth and elevation samples independently.

However, it is possible to just keep the marginal weights to not store the $N \times M$ joint weights. To do so, the weights $\omega_{\theta_{in}}$ and $\omega_{\phi_{im}}$ of both GMMs must be updated according to the current distribution:

$$p(r_i|\mathbf{x}_r^{t+1}, \mathbf{x}_i^{t+1}, \rho_i, \theta_i, \phi_i) \tag{3.41}$$

But, as can be notice, this probability depends on both, the azimuth and the elevation angle. Hence the first method proposed in this dissertation calculates the marginal weights using the following expressions:

$$\omega_{\theta_{in}}^{t+1} = \omega_{\theta_{in}}^{t} \arg\max_{\phi_{im}^t} p(r_i^{t+1}|\mathbf{x}_r^{t+1}, \mathbf{x}_i^{t+1}, \rho_i^{t+1}, \theta_{in}^{t+1}, \phi_{im}^{t+1}) \tag{3.42}$$

$$\omega_{\phi_{im}}^{t+1} = \omega_{\phi_{im}}^{t} \arg\max_{\theta_{in}^t} p(r_i^{t+1}|\mathbf{x}_r^{t+1}, \mathbf{x}_i^{t+1}, \rho_i^{t+1}, \theta_{in}^{t+1}, \phi_{im}^{t+1}) \tag{3.43}$$

These expressions take the elevation angle which maximizes the azimuth angle weight n, and the azimuth value which maximizes the weight of an elevation angle weight m. This method is referred here as Maximum Likelihood Update (MLU).

Weights update method 2: Total Probability Update (TPU)

In this second method, the marginal distributions:

$$\omega_{\theta_{in}}^{t+1} = \omega_{\theta_{in}}^{t} p(r_i^{t+1}|\mathbf{x}_r^{t+1}, \mathbf{x}_i^{t+1}, \rho_i^{t+1}, \theta_{in}^{t+1}) \tag{3.44}$$

$$\omega_{\phi_{im}}^{t+1} = \omega_{\phi_{im}}^{t} p(r_i^{t+1}|\mathbf{x}_r^{t+1}, \mathbf{x}_i^{t+1}, \rho_i^{t+1}, \phi_{im}^{t+1}) \tag{3.45}$$

are calculated from the joint conditional probability shown in (3.41) assuming an independence between landmark bearing parameters and using the Total Probability Theorem for discrete random variables [92]. This method is referenced along this chapter for comparison purpose as Total Probability Update (TPU):

$$p(r_i|\mathbf{x}_r, \mathbf{x}_{\theta_{in}}) = \sum_{m=1}^{M} p(r_i|\mathbf{x}_r, \mathbf{x}_{\theta_{in}}, \phi_{im}) p(\phi_{im}) \tag{3.46}$$

$$p(r_i|\mathbf{x}_r, \mathbf{x}_{\phi_{im}}) = \sum_{n=1}^{N} p(r_i|\mathbf{x}_r, \mathbf{x}_{\phi_{im}}, \theta_{in}) p(\theta_{in}) \tag{3.47}$$

Where $\mathbf{x}_{\theta_{in}} = [\mathbf{x}_i, \rho_i, \theta_{in}]$, $\mathbf{x}_{\phi_{im}} = [\mathbf{x}_i, \rho_i, \theta_{im}]$, $p(\theta_{in}) = \omega_{\theta_{in}}^{t}$ and $p(\phi_{im}) = \omega_{\phi_{im}}^{t}$. Conditional probabilities $p(r_i|\mathbf{x}_r, \mathbf{x}_{\theta_{in}}, \phi_{im})$ and $p(r_i|\mathbf{x}_r, \mathbf{x}_{\phi_{im}}, \theta_{in})$ are evaluated as Gaussian distributions, with mean computed with (3.31) for each hypotheses composed by θ_{in} and ϕ_{im} modes, and variance $\sigma_{r_i}^2$.

TPU has the same computational complexity of MLU. The difference with MLU, is that TPU considers the complete multi-modal distribution of the azimuth and elevation angles instead of using the most probable Gaussian mode, making the method more robust against noisy measurements. The main advantage of considering azimuth and elevation samples as independent variables is that the weights of these parameters can be stored with a storage complexity $O(N + M)$ against the storage complexity required to store each joint hypotheses weight $O(N \times M)$.

3.4.3 Inter-landmark measurements

Previous section have described the observation model employed to correct the EKF state with range-only measurements generated from the robot to one beacon (i.e. a landmark). However, as Figure 3.4 shows, there are two additional types of multi-modal range measurements that are those taken from two static range-only sensors, i.e. those generated from anchors (range-only sensors which position is

given) to beacons and those generated from one beacon to other. These inter-node range measurements are not used to initialize any landmark since it is considered that only landmarks on the operational range of the robot are of interest for computational reasons. Indeed it could be implemented a method to remove landmarks from filter when they are not observed for a long period of time in case of long-term SLAM (this thesis implemented methods to remove landmarks but is not used for this purpose since it is out of the framework application of the thesis).

In this section, for the application of inter-node measurements it is only considered the case of reduced parameterization with GMC correction model and TPU weights strategy. Then, the comparisons performed in the experimental results of this chapter between parameterizations, correction models and weights strategy will not use inter-landmark measurements.

Anchor to beacon observations

For the case of anchor-landmark range-only observations $r_{A_i u}$, the observation model of these measurements is quite similar to (3.31) but with $\delta_x = (x_{f_i} - x_{A_i})^2$, $\delta_y = (y_{f_i} - y_{A_i})^2$ and $\delta_z = (z_{f_i} - z_{A_i})^2$. Thus, in this case, as the robot state vector x_r is not used, and $\mathbf{x}_{A_i}$ is not part of the state vector, the first terms of Jacobian related with robot position (3.33) would be $\frac{\partial h(\mathbf{x})}{\partial \mathbf{x}_r} = 0$.

On the other hand, the conditional probability used to update the weights of the GMMs are no longer conditionally dependent on the robot position but on the fixed position of the anchor $\mathbf{x}_{A_i}$. Hence, the new conditional probability to be calculated in TPU is:

$$\omega_{\theta_{in}}^{t+1} = \omega_{\theta_{in}}^{t} p(r_i^{t+1} | \mathbf{x}_{A_i}, \hat{\mathbf{x}}_i^{t+1}, \rho_i^{t+1}, \theta_{in}^{t}) \tag{3.48}$$

$$\omega_{\phi_{im}}^{t+1} = \omega_{\phi_{im}}^{t} p(r_i^{t+1} | \mathbf{x}_{A_i}, \hat{\mathbf{x}}_i^{t+1}, \rho_i^{t+1}, \phi_{im}^{t}) \tag{3.49}$$

Inter-beacon observations

In the case of inter-landmark range-only observations r_{uv}, this chapter proposes to use (3.31) but with $\delta_x = (x_{f_u} - x_{f_v})^2$, $\delta_y = (y_{f_u} - y_{f_v})^2$ and $\delta_z = (z_{f_u} - z_{f_v})^2$. Thus, the Jacobian $\mathbf{H}$ of this observation model is computed using landmark Jacobian (3.35) as:

$$H = [0, \ldots, 0, H_u, 0, \ldots, H_v, 0, \ldots, 0] \tag{3.50}$$

The probability distribution functions employed to update the weights of both landmarks' GMMs are:

$$\omega_{\theta_{un}}^{t+1} = \omega_{\theta_{un}}^{t} p(r_{uv}^{t+1} | \mathbf{f}_v^{t+1}, \mathbf{x}_u^{t+1}, \rho_u^{t+1}, \theta_{un}^{t+1}) \tag{3.51}$$

$$\omega_{\phi_{um}}^{t+1} = \omega_{\phi_{um}}^{t} p(r_{uv}^{t+1} | \mathbf{f}_v^{t+1}, \mathbf{x}_u^{t+1}, \rho_u^{t+1}, \phi_{um}^{t+1}) \tag{3.52}$$

$$\omega_{\theta_{vn}}^{t+1} = \omega_{\theta_{vn}}^{t} p(r_{uv}^{t+1} | \mathbf{f}_u^{t+1}, \mathbf{x}_v^{t+1}, \rho_v^{t+1}, \theta_{vn}^{t+1}) \tag{3.53}$$

$$\omega_{\phi_{vm}}^{t+1} = \omega_{\phi_{vm}}^{t} p(r_{uv}^{t+1} | \mathbf{f}_u^{t+1}, \mathbf{x}_v^{t+1}, \rho_v^{t+1}, \phi_{vm}^{t+1}) \tag{3.54}$$

In this case, conditional probabilities of (3.51)-(3.54) are again computed using Total Probability Theorem over variables ϕ_{um}, θ_{vn} and ϕ_{vm} for (3.51), variables θ_{un}, θ_{vn} and ϕ_{vm} for (3.52), variables ϕ_{vm}, θ_{un} and ϕ_{um} for (3.53) and variables θ_{vn}, θ_{un} and ϕ_{um} for (3.54).

$$p(r_{uv}|\mathbf{x}_{\theta_{un}}) = \sum_{i=1}^{M_u} \sum_{j=1}^{N_v} \sum_{o=1}^{M_v} p(r_{uv}|\mathbf{x}_{\theta_{un}}, \phi_{ui}, \theta_{vj}, \phi_{vo}) p(\phi_{ui}) p(\theta_{vj}) p(\phi_{vo}) \tag{3.55}$$

$$p(r_{uv}|\mathbf{x}_{\phi_{um}}) = \sum_{i=1}^{N_u} \sum_{j=1}^{N_v} \sum_{o=1}^{M_v} p(r_{uv}|\mathbf{x}_{\phi_{um}}, \theta_{ui}, \theta_{vj}, \phi_{vo}) p(\theta_{ui}) p(\theta_{vj}) p(\phi_{vo}) \tag{3.56}$$

$$p(r_{uv}|\mathbf{x}_{\theta_{vn}}) = \sum_{i=1}^{N_u} \sum_{j=1}^{M_u} \sum_{o=1}^{N_v} p(r_{uv}|\mathbf{x}_{\theta_{vn}}, \theta_{ui}, \phi_{uj}, \phi_{vo}) p(\theta_{ui}) p(\phi_{uj}) p(\phi_{vo}) \tag{3.57}$$

$$p(r_{uv}|\mathbf{x}_{\phi_{vm}}) = \sum_{i=1}^{N_u} \sum_{j=1}^{M_u} \sum_{o=1}^{N_v} p(r_{uv}|\mathbf{x}_{\phi_{vm}}, \theta_{ui}, \phi_{uj}, \theta_{vo}) p(\theta_{ui}) p(\phi_{uj}) p(\theta_{vo}) \tag{3.58}$$

The update of landmarks' GMMs weights might be computationally expensive in case of inter-landmark observations when these landmarks contains a high number of hypotheses, however, in practice, as landmarks are static, this observations can be integrated in the EKF at a low frequency to avoid filter overconfidence regarding the landmarks positions. On the other hand, as will be shown during experimental validation, inter-landmark observations, together with robot-landmark measurements, make the hypotheses convergence faster. Thus, once both landmarks converge to a single hypotheses, the application of (3.51)-(3.54) is as cheap as integrating any other range-only measurement, situation in which the weights do not need to be updated.

The following subsection will describe the Gaussian Mixture reduction technique used to reduce the number of modes in Gaussian Mixtures, the method is equally applied for any kind of range measurement involving one or two landmarks.

3.4.4 GMMs Reduction

As in other multi-hypotheses methods, it is highly recommended to eliminate those hypotheses which become less probable or merge those similar. This reduction lowers the computational requirements of multi-hypotheses methods which, in the case of GMC correction, only affects to the length of the EKF state vector but not to the GMC observation model computation.

The heuristics used in this dissertation to reduce the number of modes of one GMM are based on the following rules:

- If a weight mode ω_{ij} is lower than a certain threshold δ_{th} then, mode j is pruned. The threshold used is $\delta_{th} = 10^{-11}/k$, where k is the current number of modes in the Gaussian Mixture. The use of variable k allows to use a dynamic threshold which is adapted to the current state of GMMs. The numerator of the threshold value 10^{-11} has been selected experimentally looking for a trade-off between accuracy and efficiency. Notice that a lower value of this numerator postpones the prune of modes but allows these modes to be updated more time.

- The second rule is used to merge similar modes. Two modes of the same GMM are merged if their relative arc distance $\rho_i|\alpha_{in} - \alpha_{im}|$ is below than a certain threshold δ_d (in practice $\delta_d = 0.25m$ gives good results), here α_{ij}

represents two different modes of a single Gaussian Mixture (the azimuth or the elevation GMM). Then in order to merge two similar modes, this section use the moment-preserve merge procedure explained in [88] because, as its name suggests, it preserves the overall moment of the Gaussian Mixture when merging two modes as compared to other methods [17, 35] which prune the one with less weight without preserving the overall moment.

As it might be noticed, as a consequence of the reduced parameterization proposed here, after eliminating a Gaussian mode θ_{in} from the θ_i GMM, actually M full hypotheses are pruned from the spherical shell distribution (in the same way, N full hypotheses are deleted when pruning a mode ϕ_{im}).

3.5 Simulated and real experiments

This section is aimed to validate the different contributions of the RO-SLAM algorithm presented. First, the localization accuracy will be tested for different number of anchors in order to overcome the constraints imposed by a noisy odometry. In this case, due to a lack of a good odometry model for the aerial robot, all experiments were performed with a Random Walk prediction model which consist on adding a fixed noise to the state vector of the robot. To test the scalability and accuracy of the proposed algorithm, different simulations for up to 50 range-only sensors will be used. This simulations will compare not only the proposed state vector, observations models and weights update strategies, but also the state-of-the-art methods for undelayed RO-SLAM with respect the approaches proposed in this thesis. The experiments will compare the algorithm when inter-landmark observations are included.

In this case, all implementations of the different methods are programmed in the C++ language under the Robot Operating System (ROS) framework [86]. Range measurements in either simulation and real experiments are typically limited to a maximum distance as the number of outliers increase linearly with the distance between the range-only sensors. On the other hand, the initialization of beacons is usually limited to measured distances below 10 meters in order to reduce the number of hypotheses/parameters in the state vector.

The range-only sensors simulated and the ones used during real experiments are those described in chapter 2 section 2.5.1.

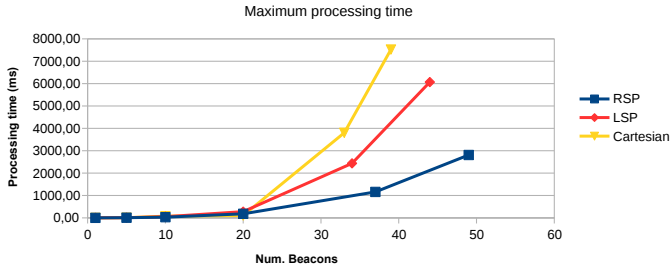
3.5.1 Simulations

Simulation results are mainly used for comparison purpose between the proposed methods and others in the literature. First experiments, compare the scalability and accuracy of the proposed reduced spherical parameterization (or RSP) against other classical parameterizations. Later, the GMC observation model is compared with other models based on Federated Information Sharing including MHC porposed here. The independence assumption made between azimuth and elevation parameters in MHU and TPU strategies will be validated by comparing the results with the classical joint hypotheses FHU strategy where azimuth and elevation parameters are considered as dependent random variables. As the main objective of these experiments is to evaluate the mapping algorithm proposed for RO-SLAM, during simulation experiments the localization of the robot is given to avoid mapping errors coming from a bad localization of the aerial robot. Real experiments will evaluate the complete SLAM approach showing localization and mapping results.

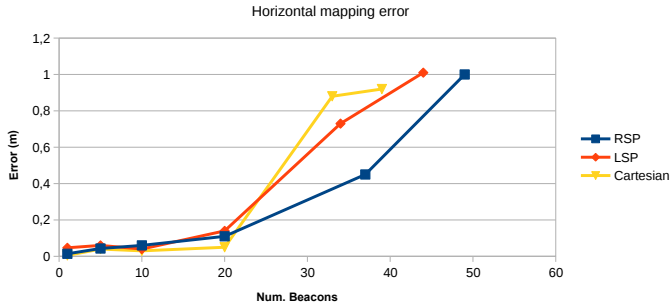
Map scalability

The scalability of the mapping approach is validated using different simulations for up to 50 landmarks randomly distributed in a region of 30X30X10 meters. The results compare the computational complexity and accuracy of the mapping approaches using large spherical parameterization (LSP) extended from [17], the 3D Cartesian parameterization used in [8] and the reduced spherical parameterization (RSP) proposed in this dissertation.

In Figure 3.3 shown above, it was demonstrated a huge improvement in the amount of parameters required to estimate the same number of hypotheses with respect other classical EKF representations and, as stated in [92], it is well known that the computational complexity of the EKF framework is directly related with the number of parameters, thus the reduction of the state vector implies an improvement in the EKF-SLAM framework. This theoretical results on computational complexity are validated with the simulation results shown in Figure 3.6a. This figure shows the maximum processing time spent during the correction stage of the EKF, i.e. during computation of correction equations in Algorithm 9. The maximum processing time is used since in most of the cases will coincide with the case of a maximum number of parameters in the state vector. On the other hand, Figure 3.6b shows the average mapping error for different



(a)



(b)

Figure 3.6: The results compare the processing time and mapping error for the proposed Reduced Spherical Parameterization (RSP), the Large Spherical Parameterization (LSP) and the classical Cartesian parameterization: (a) EKF correction stage maximum processing time, (b) averaged mapping error.

number of beacons. As can be seen, the mapping error is approximately the same for all parameterizations but with clearly reduced computational complexity in the case of the reduced parameterization due to its reduced number of parameters shown in Figure 3.3. Another characteristic of the mapping error is the increasing mapping error with the number of landmarks which is mainly related with a reduced convergence time of the solution due to the inter-landmark correlations introduced by EKF.

In this case, the scalability is studied with the number of landmarks (or beacons). However, the same results might be reached by using just one beacon

and increasing the initialization distance or the density of hypotheses d.

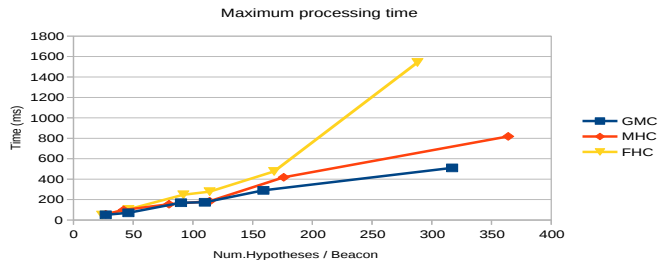
Observation model

The following simulations are aimed to compare the classical observation model (FHC), with one equation per joint hypotheses, and the observation models proposed in this chapter. For all these observation models the proposed reduced spherical parameterization is used to demonstrate how this parameterization fits also with classical observation models. In this case, the results are compared against different initialization densities d which implies different number of hypotheses per beacon and with a fixed number of 10 beacons. The idea is to compare the processing time with a larger number of correction equations in the observation model and also with a different number of parameters in the state vector.

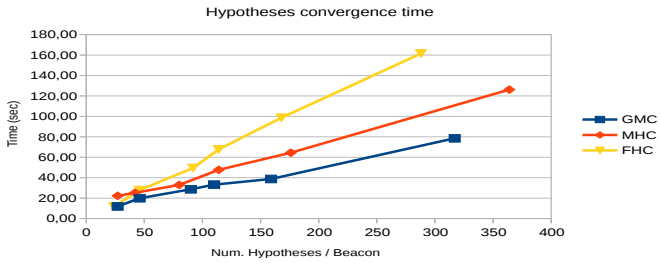
As shown in line 3 of Algorithm 9 used to update the covariance of the EKF, the computational burden on the inversion of the innovation matrix S depends on the number of correction equations (i.e. number of rows of observation model Jacobian H), whereas the rest of equations of the EKF depends on the number of parameters in the state vector. Figure 3.7a shows the processing time taken not only in the correction stage of the EKF but also in computing the matrices of the range-only observation model (Jacobian, noise matrix, etc). The figure shows a quadratic increment for FHC due to the quadratic dependence with the number of azimuth and elevation angles while the other models are more linear. In the case of GMC the linear increment on the processing time is because of the increment in the number of parameters on the state vector. MHC depends not only on the number of parameters to model all hypotheses but in the number of correction equations used. On the other hand, Figure 3.7b shows a faster convergence time to reach a single hypothesis in GMC with respect the other models which comes at expense of a higher mapping error as shown in Figure 3.7c. As this last figure shows, the mapping error decreases as the density of hypotheses increments during initialization phase. This decrement is lower for FHC and higher for MHC.

Weights update

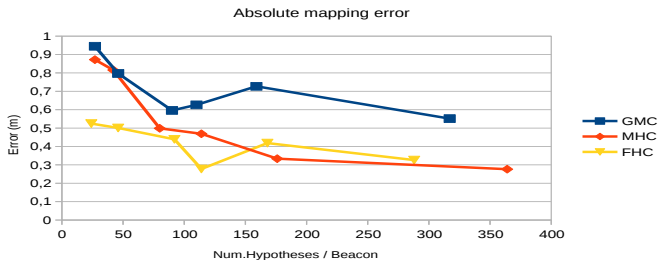
Despite the convergence time of beacons depends in part on the observation model implemented (FHC, MHC or GMC), it is mainly related with the strategy



(a)



(b)



(c)

Figure 3.7: The results compare (a) the processing time, (b) hypotheses convergence time and (c) averaged absolute mapping error for 10 beacons, different number of hypotheses and different range-only observation models.

used to update the weights of GMMs. Correction strategies MHC and GMC can be combined with different methods used to update the weights of GMMs (e.g. MHC with MLU or MHC with TPU, etc). However, as MLU and TPU are

based on an independence assumption between azimuth and elevation angles, the classical Cartesian and LSP parameterizations are not suitable for these update strategies. On the other hand, as the reduced parameterization does not impose an independence between beacon parameters, this parameterization and the GMC observation model proposed here can be still used with the classical approach used to update the weights of hypotheses at expenses of a high memory consumption. The following experiments compare classical Full Hypotheses Update (FHU) strategy used in the literature for dependent azimuth and elevation parameters with the MLU and TPU methods proposed in this chapter, Most Likely Update (MLU) and Total Probability Update (TPU). In these experiments 10 beacons are used and the observation model used in these experiments is GMC.

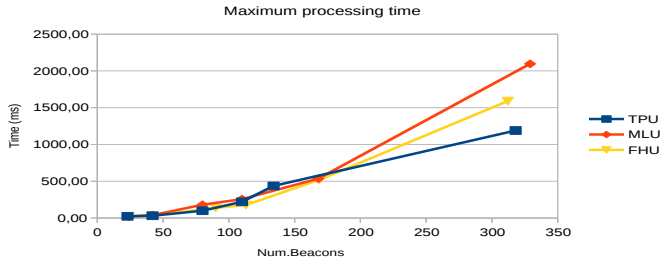
In this case, the results are also compared against different hypotheses densities d to check the update processing time and the hypotheses convergence for the different strategies with a larger number of hypotheses per beacon. The processing time measured is the time used to update the weights of the Gaussian Mixtures plus the processing time taken by the prune strategy explained above.

As Figure 3.8 shows there is not much difference between FHU and TPU methods in terms of accuracy and both have a better performance than MLU. However, TPU tends to be the most efficient in terms of processing time and memory consumption. FHU requires a quadratic memory consumption with the number of azimuth and elevation angles as explained above due to the dependence assumption made between landmark parameters. Additionally, the independence assumption does not seem to affect the accuracy of the mapping results when using either MLU or TPU.

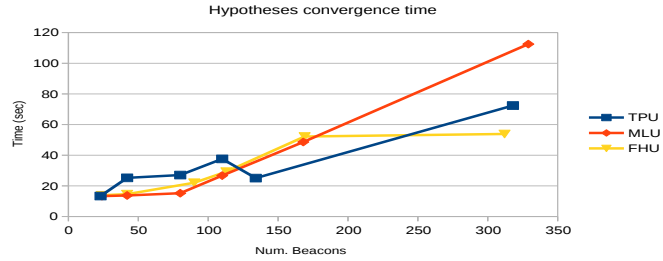
Inter-node range-only observations

This section validates the observation model for inter-node range-only observations in the EKF-SLAM framework described above. The same synthetic used before is reused now but integrating inter-landmark observations. Inter-landmark measurements r_{ij} are filtered so that they are integrated in the filter at a frequency of 0.1Hz for the same pair of sensors i and j. This criteria is necessary to reduce redundant information which can lead to overconfident estimations. In this case, the method employed for correction stage and to update the weights of hypotheses are GMC and TPU respectively.

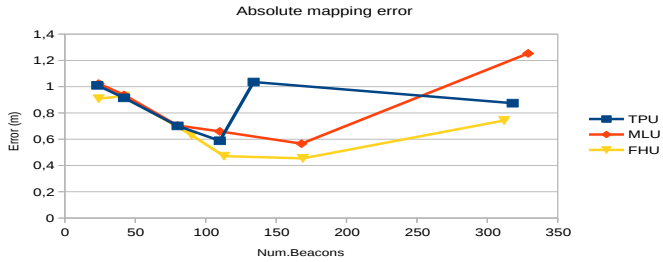
The localization error using inter-node observations in the case of the synthetic dataset for 20 beacons goes from 0.63 meters to 0.49 meters. This reduction in



(a)



(b)



(c)

Figure 3.8: The results compare (a) the processing time, (b) hypotheses convergence time and (c) averaged absolute mapping error for 10 beacons, different number of hypotheses and different weights update strategies.

the localization error is a direct consequence of the reduction on the mapping errors but also in the convergence time. It should be noticed that, while a beacon does not converge, the corrections introduced by the range observations of these

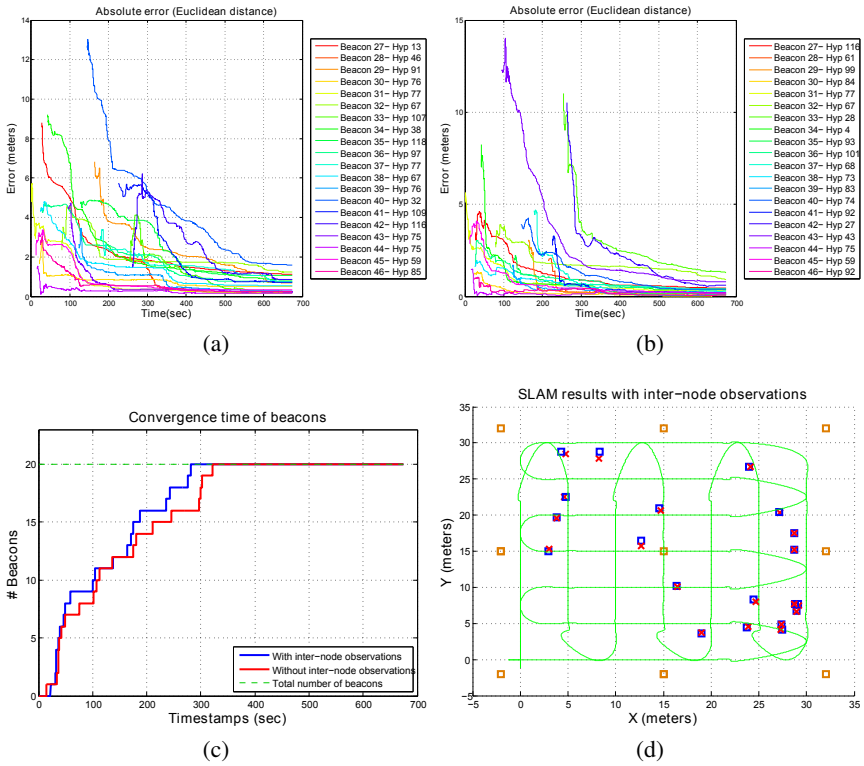


Figure 3.9: Experiments with inter-node range measurements: (a) simulation with 20 beacons and without inter-node observations, (b) simulation with 20 beacons and inter-node observations, (c) convergence time of beacons with (blue line) and without (red line) inter-node observations and (d) shows the error of the mapping error using inter-landmark observations in 2D view.

beacon are less precise due to its multi-hypotheses state.

On the other hand, in Figure 3.9 can be seen how inter-node range measurements reduces the mapping error in more than 45%. The convergence time is also reduced with this inter-node observations as shown in Figure 3.9c in more than 15%. The X axis represents the time stamps of the experiment and the Y axis the number of beacons which have converged to a single hypotheses. The blue line correspond to the convergence time using inter-node range measurements whereas the red line corresponds to the convergence time without using inter-node

range measurements. This reduction of the convergence time is dependent on the frequency at which inter-node observations are integrated in the filter. Simulations performed at different inter-node observation periods have shown that higher frequencies leads to faster convergences but may reduce filter stability.

3.5.2 Real experiments

During all experiments, only aerial vehicles were considered to validate the 3D RO-SLAM algorithm. The main problem of aerial robots is that it is hard to obtain a precise odometry. Thus, in these experiments, in order to localize the robot when the map is not initialized, a set of static sensor nodes with known position (anchors) are used to localize the robot. Some aerial robots use visual odometry to cope with this problem. In this thesis, anchors allows the aerial robot to be localized by multilateration of the robot position by using range measurements taken from the robot to anchors. Despite a precise prediction model might improve the results of the experiments presented in this dissertation, a simple prediction model has been implemented for these experiments which just increases the variance of robot parameters with a very small variance value at a high frequency. Real experiments are based on real datasets that have been made public at the end of this thesis for the research community. The details of the setup employed on the different experiments are detailed in appendix B.

Localization experiments

In the case of 3D localization at least 4 anchors are required to estimate the three-dimensional position of the robot. In this case the localization algorithm is tested using the indoor setup detailed in appendix B section B.4. In order to test the accuracy of the robot localization (not SLAM) with respect the number of anchors, different experiments with a real aerial vehicle in an indoor environment (see Figure B.5) were performed with up to 16 anchors (see Figure B.3a).

To complement the poor trilateration of the aerial robot altitude, the on-board altimeter of the Pelican was used, including the estimation of the altimeter bias in the EKF. The results of the robot localization with and without altimeter are depicted in Figure 3.10.

The figure shows the absolute localization error computed with the euclidean distance between the estimated robot position from the EKF and the robot ground-truth. As can be seen, without altimeter (red line), the localization error of the

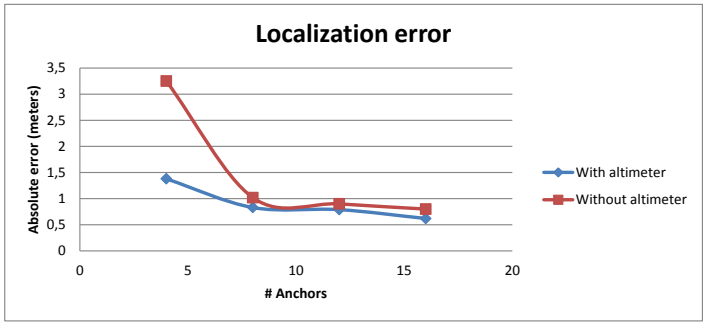


Figure 3.10: Localization error with respect the number of anchors employed. This figure shows the results of the robot localization for different number of anchors where the blue line is the localization fusing range measurements with an altimeter and the red line is the results without using the altimeter.

aerial robot is higher due to a bad trilateration of the robot altitude. The bad trilateration is a consequence of the difference on the horizontal trajectory of the robot in the XY plane with respect to the trajectory along axis Z. The results are improved fusing range-only with altimeter measurements (blue line). As this figure shows, the localization error is virtually the same ($\leq 1m$) when using 8 or more anchors. The horizontal and vertical localization error as well as the absolute localization error are summarized in Table 3.2. This table shows how the absolute error is reduced as the number of anchors increases. In the following experiments 8 anchors will always be used fused with the barometer.

SLAM results

For real experimentation an ARCAS set-up was used, where 6 beacons were embedded in 3 bars (2 beacons per bar) as shown in Figure B.3c. In this experiment, only MHC and GMC observation models are compared using MHC with MLU as the RO-SLAM algorithm presented in [35] against the algorithm proposed in this chapter using GMC with TPU. The results obtained with this real dataset are shown in Figure 3.12. The averaged localization error is 0.54 meters and the averaged mapping error is 0.6 meters with an averaged horizontal error of 0.14 meters for the MHC method, whereas for the GMC method the localization error is 0.54 meters and the averaged mapping error is 0.58 meters with an averaged horizontal error of 0.2 meters. Figure 3.11a shows the localization

Table 3.2: Localization error using different number of anchors and an altimeter.

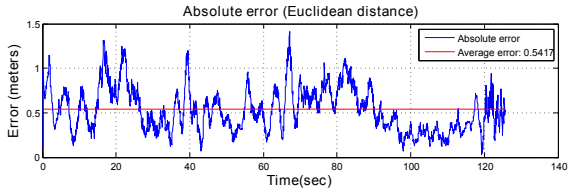
	Errors (meters)			
#Anchors	**Abs.**	**Horizontal**	**Vertical**	**Altimeter**
4	1.51	1.5	0.22	Used
8	1.03	1.0	0.21	Used
12	0.79	0.75	0.22	Used
16	0.62	0.59	0.22	Used
4	3.25	1.53	2.8	Not used
8	1.02	0.78	0.65	Not used
12	0.9	0.76	0.75	Not used
16	0.8	0.59	0.7	Not used

results for GMC (the MHC results are similar). As shown in Figure 3.13 with a red line, the 75% of times the localization error is 0.6 meters when not using inter-node range-only observations. Furthermore, as Figure 3.11b shows with a green dashed line, the localization estimation shown in blue is always within the 3σ variance interval with respect the ground-truth of the aerial robot shown with a red line.

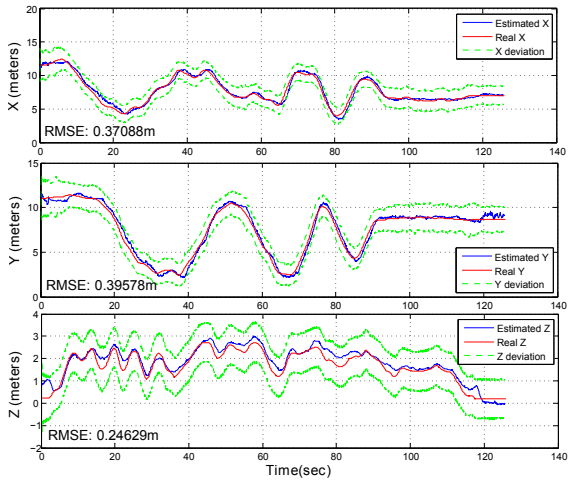
Mapping error is shown in Figure 3.12a for MHC and in Figure 3.12b for GMC. In this case, as it was an indoor experiment, the aerial robot only could flight in a range of 1.5-3 meters the reason why the mapping error presents a higher vertical mapping error which is specially notable in the case of beacon 22 because of a lack of observability in the beacon altitude. Again, the SLAM results are quite similar in both methods with a mapping error below 0.5 meters but with a reduced computational complexity in the case of GMC.

Inter-node range-only observations

This section validates the observation model for inter-node range-only observations in the EKF-SLAM framework using the same real dataset used above. Inter-landmark measurements r_{ij} are filtered so that they are integrated in the filter at a frequency of 0.1Hz for the same pair of sensors i and j. This criteria is necessary to reduce redundant information which can lead to overconfident estimations. In this case, the method employed for correction stage and to update the weights of hypotheses are GMC and TPU respectively.



(a)



(b)

Figure 3.11: Localization results for real dataset: (a) absolute localization error in blue and RMS error in red using GMC, (b) localization error with GMC along X, Y and Z axis. In (b) estimations are the blue continuous lines, the ground-truth is drawn with a red continuous line and the 3σ variance interval is drawn with a green dashed line. Localization errors are virtually the same for MHC.

The localization error using inter-node observations in the case of real experiments goes from 0.54 meters to 0.49 meters. In the cumulative localization error shown in Figure 3.13 it can be seen how the localization error throughout the whole experiment is reduced when inter-node range measurements are used. This reduction in the localization error is a direct consequence of the reduction on the mapping errors but also in the convergence time.

On the other hand, in Figure 3.14 can be seen how inter-node range measurements reduces the mapping error in more than 6% for real experimentation. The convergence time is also reduced with this inter-node observations as shown in

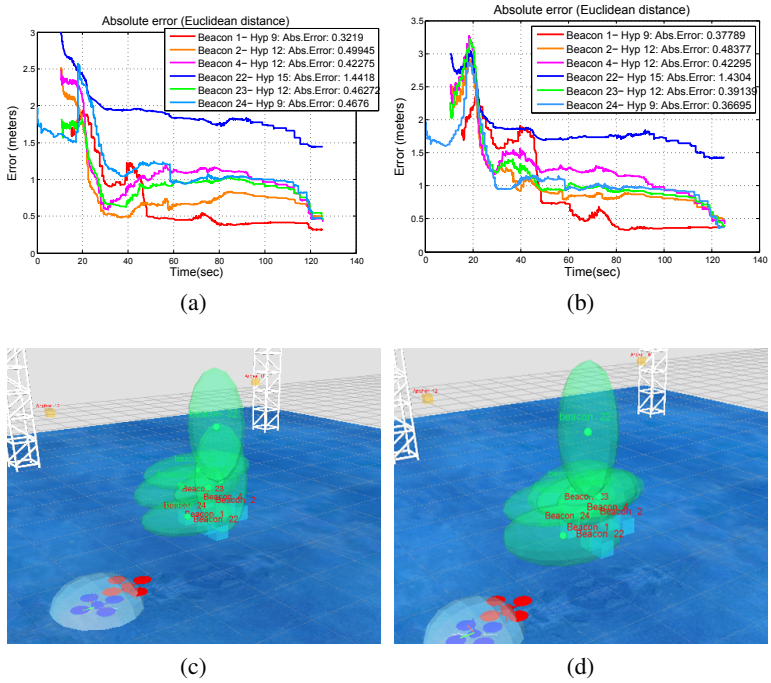


Figure 3.12: Mapping results for real dataset: (a) mapping results with MHC method and (b) mapping results with GMC method. (c) shows the 3D view of the mapping results with MHC and (c) shows the 3D view of the mapping results with GMC.

Figure 3.14c in more than 60% in real experimentation. The X axis represents the time stamps of the experiment and the Y axis the number of beacons which have converged to a single hypotheses. The blue line correspond to the convergence time using inter-node range measurements whereas the red line corresponds to the convergence time without using inter-node range measurements. This reduction of the convergence time is dependent on the frequency at which inter-node observations are integrated in the filter, simulations performed at different inter-node observation periods have shown that higher frequencies leads to faster convergences but may reduce filter stability.

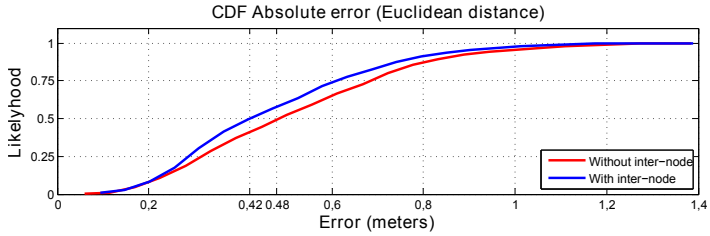


Figure 3.13: Cumulative distribution function of localization errors with (blue line) and without (blue line) inter-node range measurements for real dataset.

3.6 Summary and conclusions

This chapter presented a multi-hypotheses SLAM approach based on the integration of Gaussian Mixtures in a EKF with undelayed initialization. The approach has been validated with Range-only SLAM which is a SLAM problem based on the single integration of multi-modal range measurements, i.e. low-informative observation composed by the distance between two range-only sensors. This observations leads to multi-modal distributions due to the lack of bearing information.

The Range-only SLAM algorithm proposed is based on a Reduced Parametrization. The chapter showed two new observations models for the correction stage of the EKF. The first model is based on Federated Information Sharing but reducing the amount of equations required and hence improving the efficiency of the correction stage. The second is based on a novel technique which only requires a single correction equation to update the multiple hypotheses in the state vector by using the expectation statistic of Gaussian Mixtures. This efficient correction scheme used for range observations not only improves the computational requirements of the correction stage but also reduces the convergence time of hypotheses and introduce a higher information gain with each range observations as compared with other multi-hypotheses schemes. Furthermore the chapter presents two new schemes to update the weights of hypotheses which have two main advantages: on the one hand it reduces the amount of memory required to store weights information compared with classical approaches and, on the other hand, enhances the convergence and robustness of other state of the art techniques. Additionally, the method uses an hypotheses prune/merge strategy which reduces the number of hypotheses in the state vector as they become less probable and hence reducing

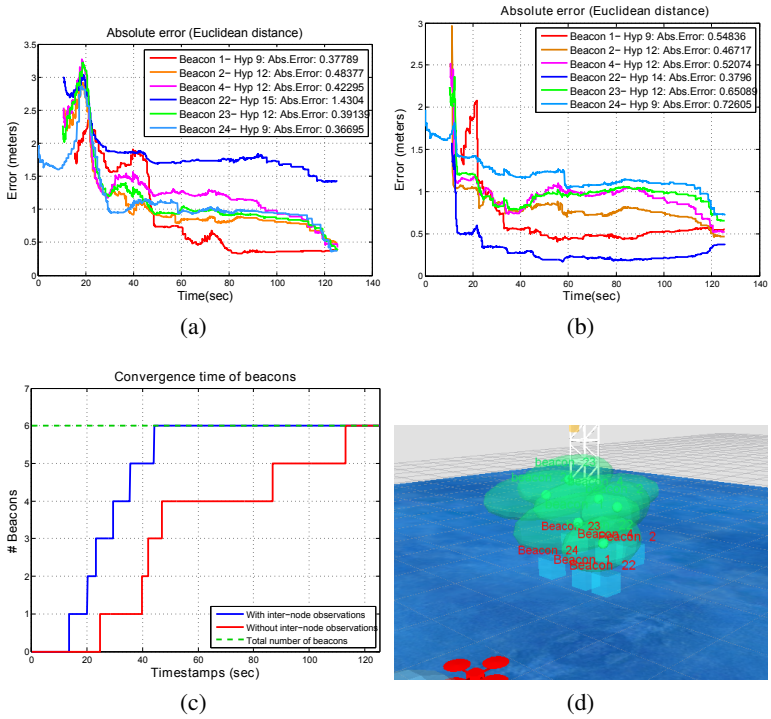


Figure 3.14: Experiments with inter-node range measurements: (a) real experiment without inter-node observations, (b) real experiment with inter-node observations, (c) convergence time of beacons with (blue line) and without (red line) inter-node observations in real experimentation and (d) shows a 3D view of the final results using Rviz from ROS framework.

the computational requirements of the algorithm. Finally, this chapter proposes a method to integrate not only robot to sensors range observations but also range observations between static range-only sensors, allowing to the reduce the convergence time of hypotheses and to refine the mapping and localization errors in up to a 60%.

The method was validated with simulated and real experiments using a real aerial robot comparing the results of the methods proposed with respect the other state of the art approaches based on undelayed EKF. The results compares the proposed reduced parameterization with classical parameterizations showing that

the proposed parameterization offers a similar precision but with a considerably better computational memory and time performance. As the results show, both correction methods presents a similar accuracy but, in the case of the second method, it increases its robustness and computational requirements by just using one correction equation and including the weights of the Gaussian Mixture Models in the correction equation. The weights strategy based on Total Probability Update also presents similar results but requiring less computational resources with respect the first method proposed and other state of the art strategies.

Next chapter will deal with sensor data filtering in order to remove outliers and a small extension of the method presented above in order to consider the estimation of the propagation model of every range-only sensor into the SLAM framework.

Chapter 4

Towards robust 3D RO-SLAM estimation

4.1 Introduction

As previously introduced, it is well known that outliers and distorted measurements significantly affects range-only simultaneous and mapping approaches [24], making them less applicable in real application scenarios. This chapter proposes improving the multi-hypothesis approach presented in previous chapter with specially adapted pre-filtering algorithms for range-only sensors in order to elevate the approach to a new level in terms of accuracy and robustness. This pre-filtering algorithms are mainly designed to avoid outliers, a task which is specially critical during the initialization stage of the map landmarks. EKF filter is quite sensitive to outliers. However, during the initialization stage, the update of GMMs weights is based on the probability of the received measurements, making this stage specially sensitive to this bad observations since they might derivate in a wrong prune of GMs modes.

Additionally, the chapter will propose estimating the propagation model associated to each radio beacon in order to account for the bias of this range-only measurements coming from the installation particularities and sensor characteristics of each radio beacon considered into the system. In addition a method for outlier rejection will be presented here.

4.2 Overview and related work

Chapter 3 proposed an undelayed 3D RO-SLAM solution based on a EKF-SLAM framework which allows the integration of inter-sensor measurements considering the correlations between them. The main drawback of this solution is that, since it is based on an EKF, the solution is very sensitive to the presence of measurement outliers or biased measurements.

In this chapter two contributions are given: first, a robust pre-filtering algorithm is proposed to avoid the divergence of the EKF filter due to outliers. Second, with radio-based sensors, it is common to find some linear errors related with the propagation model of each node which is affected by the environment where these are placed causing biased range-only measurements. This chapter proposes a method to estimate the range measurement model parameters which corrects this linear errors, allowing then a better localization of an UAV with the results of the mapping process.

In the field of robust range-only localization an mapping [80] proposed a robust outlier rejection method to improve the performance of its EKF-SLAM based on a spectral graph partitioning. The main problem of this solution is that, as in [76], it requires a minimum network connectivity constraint to make it work properly which is not always available in real applications. The solution proposed in [76] is based on what they call *robust quadrilaterals* to avoid the ambiguities related with the multi-modal nature of range measurements. In [28] the authors propose another solution based on gating techniques, such as chi-square filter. Other authors propose [100] to use the estimated range measurement probability distribution from the EKF parameters to detect the outliers.

The proposed pre-filtering method in this chapter is based on some heuristics which depends on the current state of the UAV position and the previous range measurements received. The second contribution of this chapter is an extension of the range-only observation model to estimate the parameters of the propagation model of radio-based sensors. For this, it will be assumed that range-only measurements are biased and scaled by unknown parameters given by the environment, the multi-hypotheses approach will be updated with new information in the EKF state vector in order to refine not only the position of the range-only sensors but also their radio propagation characteristic.

4.3 Range-only measurements pre-filtering

As previously shown in Figure 3.4, the first stage executed upon receiving new measurements is the pre-filtering stage. The prefiltering method implemented in this thesis has as a main goal making the RO-SLAM algoritm more robust against outliers. The main pre-filtering algorithm is composed by a chain of filters. Each phase of the pre-filtering algorithm might reject or modify the received range measurements. The complete filter chain is summarized in Figure 4.1.

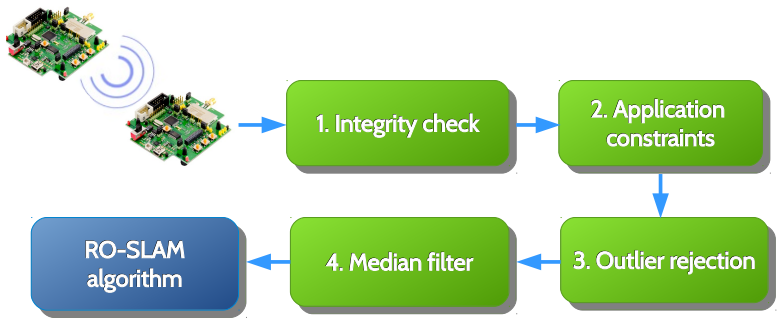


Figure 4.1: Pre-filtering phases used to improve the robustness of the RO-SLAM algorithm against noisy range-only measurements, outliers or inconsistencies. The pre-filtering algorithm proposed is based on 4 efficient filtering phases. Each phase modifies or rejects a range measurement according to specific goals of each filtering algorithm.

The following sections will describe the details of each filtering phase of the chain.

4.3.1 Integrity check

The first pre-filtering stage is in charge of rejecting those range-only measurements which are not consistent with current system state. Thus, for example, as each measurement needs a certain amount of processing time by EKF, this phase will be in charge of rejecting those range-only measurements which time-stamp is older than a certain threshold with respect current time-stamp. Other example to be rejected are those measurements which were taken from two anchors (i.e. not anchor to robot or anchor to beacon). This measurements, between anchors are rejected since they not introduce any kind of information to the RO-SLAM algorithm. This first stage will also reject range measurements which have an inconsistent value like negative or infinite values due to errors on the range-only sensor driver.

Range-only measurements classified as consistent are propagated to the next pre-filtering stage.

4.3.2 Application constraints

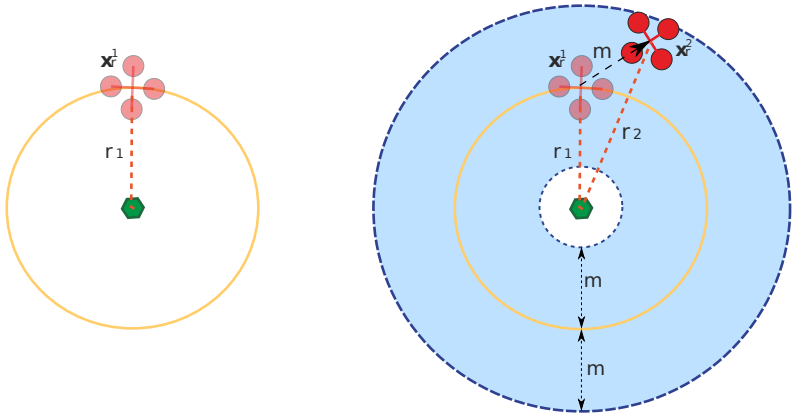
The next stage of the pre-filtering algorithm is in charge of rejecting those range-only measurements which does not satisfy a specific RO-SLAM application constraints. Thus, the rules to be checked on this filter might vary depending on the application. The rules proposed here are common rules which might be found on any application.

Common constraints for this stage consist on rejecting those range measurements which comes from an unknown sensor ID, or rejecting range measurements with a value higher or lower than a certain distance threshold or even rejecting beacon or anchors range measurements depending on the application. For example, beacon range observations should be rejected when running a localization algorithm since beacon range-only measurements are only used to map their position but not for those applications which only requires a localization of the mobile robot based on anchors. A possible implementation of this pre-filtering stage will be evaluated during experimental results where, in this case the RO-SLAM application, rejects range measurements over a certain threshold only for the initialization phase of a beacon in order to reduce the computational requirements of having a lot of hypotheses[1].

4.3.3 Outlier rejection

The next stage consist of an outlier detection and rejection algorithm. This filter is based on the current UAV position estimation. The outlier detector algorithm registers the last pose estimation of the UAV $\mathbf{x}_r^t$ every time a new range measurement r_i^t is received. Then, when new range-only measurements are received from the same pair of devices, the algorithm computes the module of the motion vector of the UAV by comparing the positions registered for the current and previous range-only measurements using the Euclidean distance, i.e. $m = ||\mathbf{x}_r^t - \mathbf{x}_r^{t-1}||$. The distance is compared with the difference between the new and last range-only measurements received $\Delta = |r_i^t - r_i^{t-1}|$. With this

[1]The number of hypotheses depends on the first range measurement received to initialize a beacon, the larger is this value r_i, the larger is the amount of hypotheses generated as shown in (3.10)



(a) (b)

Figure 4.2: First step of the outlier rejection filter. Figure (a) represents the instant of the first range measurement reception whereas (b) represent the time-stamp when a new range-only observation is received. The black arrow represents the real direction of the vehicle and the blue discontinuous line represents the radial trajectory with respect the beacon position. In this case the distance r_i^t is lower than $r_i^{t-1} + m_t + \sigma_{r_i^t}$ so the range measurement is valid.

information, the algorithm rejects all range measurements which do not follow the following constraint:

$$r_i^{t-1} - m - \sigma_{r_i^t} \leq r_i^t \geq r_i^{t-1} + m + \sigma_{r_i^t} \tag{4.1}$$

i.e. the position difference m represents the maximum distance increment allowed with an additive confidence interval of $2\sigma_{r_i^{t-1}}$. As shown in Figure 4.2a and Figure 4.2b, the module of the motion vector represents the bounds of the distance increment that might be sensed between the new UAV position with respect a range-only sensor. These bounds represents a movement in the radial direction of the range-only sensor sensed from the UAV as depicted in Figure 4.2.

4.3.4 Median filter

Once all range-only outliers have been rejected, the last pre-filtering stage consist on the application of a median and mean moving filter. This filter considers a moving window of size λ which is based on the time-stamp of measurements

and on the position of the robot where they were received. Past filtered and new range-only observations inside this window are used to compute the median value $\hat{r}_i$ of these range-only observations. Hence, range-only measurements inside this window are ordered so that the median value should fall in the middle of the ordered list of measurements of size l. Once the median value has been computed the algorithm selects an interval around the median range measurement $\hat{r}_i$ of size ωl, where ω is a percentage threshold over the measurements inside the window which value goes from 0 to 1. All measurements inside the predefined interval around $\hat{r}_i$ are used to compute the mean value of the interval $\bar{r}_i$ which is the final range measurement returned to the RO-SLAM algorithm.

The last two stages of this filter chain are summarized in Algorithm 3.

Algorithm 3: Outlier rejection and median filter

Data: r_i^t, $\sigma_{r_i^t}$ and $\mathbf{x}_r^t$
Result: Filtered range measurement $\bar{r}_i^t$

```
begin
    // Outlier rejection
    m ⟵ EuclideanDistance(x_r^{t-1}, x_r^t);
    if r_i^t > r_i^{t-1} + m + σ_{r_i^t} or r_i^t < r_i^{t-1} − m − σ_{r_i^t} then
        Discard r_i^t //Is an outlier
    // Median filter
    r_l ⟵ r_i sensed further than λ meters with respect x_r^t;
    r'_l ⟵ Order(r_l);
    r̂_i^t ⟵ Median(r'_l);
    r''_l ⟵ ValuesAround(r̂_i^t, r'_l, ω);
    r̄_i^t ⟵ Mean(r''_l);
```

4.4 RO-SLAM with propagation model estimation

It is very common that radio-based range-only sensors produce biased and, at some point, scaled measurements. This effect is the combination of the sensors environment (position, structure, ...) and the inaccuracies of the ranging mechanism. This is why both, bias and scale factor, are different for every range-only sensor in

the system. Thus, an accurate range-only localization will not only depend on the anchors position but also on the characterization of their measurement models.

One of the main advantages of the proposed 3D RO-SLAM with undelayed initialization is that the common parameters to all position hypotheses are not repeated into the filter, producing a dramatic reduction in the state vector size as shown in chapter 3. Thus, although a multi-hypotheses scheme is used, common variables such as the center of the sphere or the radius are instantiated just once per landmark into the EKF state vector, so the updating information is not shared as happens with θ and ϕ values. This section proposes estimating the measurement model bias b_i and scale factor s_i for each sensor node i into the system. This way, the feature descriptor of (3.4) will include these two parameters, obtaining the following new parameterization of sensor node in the filter:

$$\mathbf{f}_i = [\mathbf{x}_i^t, \rho_i, \theta_{i1}, \theta_{i2}, \ldots, \theta_{in_\theta}, \phi_{i1}, \phi_{i2}, \ldots, \phi_{in_\phi}, s_i, b_i]^T \tag{4.2}$$

Notice how it is included only two new parameters and how they are not affected at all by the multi-hypotheses model associated to the sensor position, leading to a fast and clean convergence of the estimation even in the presence of many hypotheses. In addition, the total overhead added to the filter is two parameters per sensor, obtaining a eight state parameterization when the feature converges to a single hypotheses[2].

With this change on the parameterization, the correction equation has been extended according to the new biased range-only measurement model. For this measurements, the following correction equation is proposed:

$$r_i = s_i\sqrt{(x_{f_i} - x_r)^2 + (y_{f_i} - y_r)^2 + (z_{f_i} - z_r)^2} + b_i \tag{4.3}$$

where x_{f_i}, y_{f_i} and z_{f_i} are defined in (3.32). Hence, the associated Jacobian $\mathbf{H}'$ of the new correction equation (4.3) is also extended with respect the landmark Jacobian (3.35) as

$$\mathbf{H}' = [s_i\mathbf{H} \quad \mathbf{h} \quad \mathbf{1}] \tag{4.4}$$

[2]To reduce this number of parameters in convergence state, it is possible to switch from a spherical to a Cartesian parameterization once there is just one landmark hypotheses, reducing the vector state from 6 to 3 parameter + 2 parameters from the propagation model (i.e. a total number of 5 parameters in Cartesian vs 8 in spherical). However, it is recommended to maintain a spherical parameterization in RO-SLAM due to the intrinsic nature of range-only measurements as already explained in [24].

where $\mathbf{h}$ is the estimated measurement obtained from (3.31).

In general, the scale factor will be initialized to 1 while the initial bias be to 0 when no fixed bias is detected during characterization. This initial values will have an associated small noise to let the filter adjust the real values of the parameters. Remember that the objective of this parameterization is to account for the small deviations of the range measurement model in commercial sensors, so the scaled factor and bias are expected to not contain a huge standard deviation.

4.5 Experimental results

The experimental setup used to validate both approaches in this chapter for real experiments is the indoor and outdoor setups detailed in Appendix B.

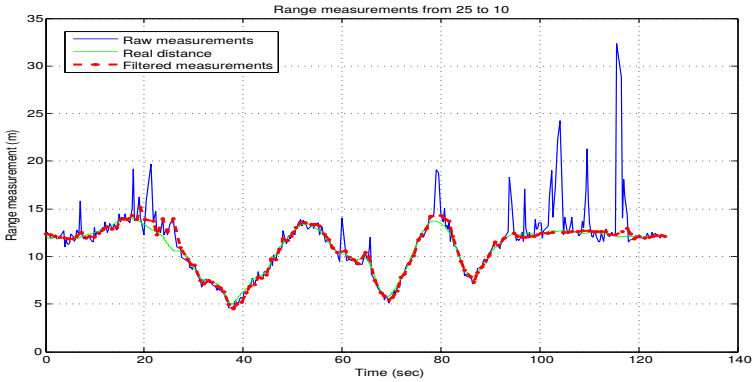
The outlier rejection algorithm is validated using the ground truth of the nodes and the aerial robot during the experiment, so the range ground truth can be easily computed using Euclidean distance. However, validating the new formulation with propagation model needs range-only sensors with biased/scaled measurements and the ground-truth. So the range information of some beacons have been artificially modified according to a known propagation model in order to see if the filter properly estimates the parameters.

4.5.1 Pre-filtering validation

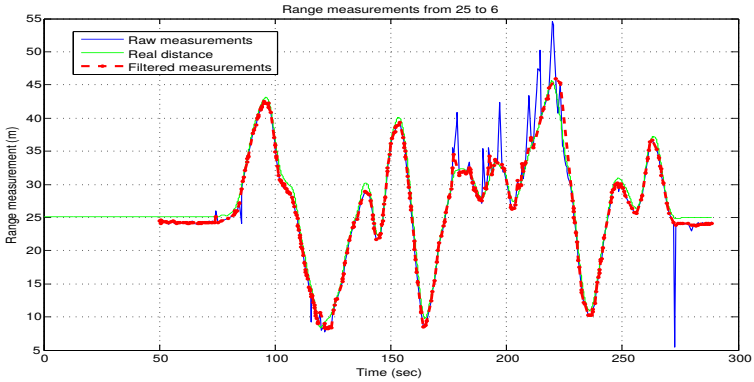
The first experiments consisted on determining whether the outlier rejection filter is working properly or not. Figure 4.3 shows two results of the outlier filter over the noisy range measurement between the UAV and one of the beacons. It can be seen how the filter removes all outliers and also produces a low-pass filtering over the range data thanks to the median filter stage reducing the effect of the noise.

Indoor validation

In order to evaluate the impact of the pre-filtering algorithm on the overall performance of the RO-SLAM approach two different experiments were carry out using one aerial robot and 6 radio-based range-only sensors which position is initially unknown. In the first experiment the RO-SLAM algorithm was executed without using the pre-filtering algorithm. In Figure 4.4 are shown the localization results of the RO-SLAM algorithm with and without using the pre-filtering algorithm.



(a) Indoor experiments



(b) Outdoor validation

Figure 4.3: Log of the range-only measurements received from the UAV to a static range-only sensor for indoor (a) and outdoor (b) experiments.The green solid line represents the real distance from the UAV to the static range-only sensor, whereas the blue solid line represents the raw measurements received from Nanotron sensor and the red solid line represents the filtered range measurement.

On the other hand, for the mapping results, Table 4.1 shows a comparison of the mapping results for the indoor real experiments with and without the use of the pre-filtering algorithm proposed in this chapter. As this table shows the pre-filtering algorithm makes the algorithm more accurate and robust against range observation outliers. This table also shows how the vertical accuracy is lower than the horizontal one. This is due to the poor trilateration performed in

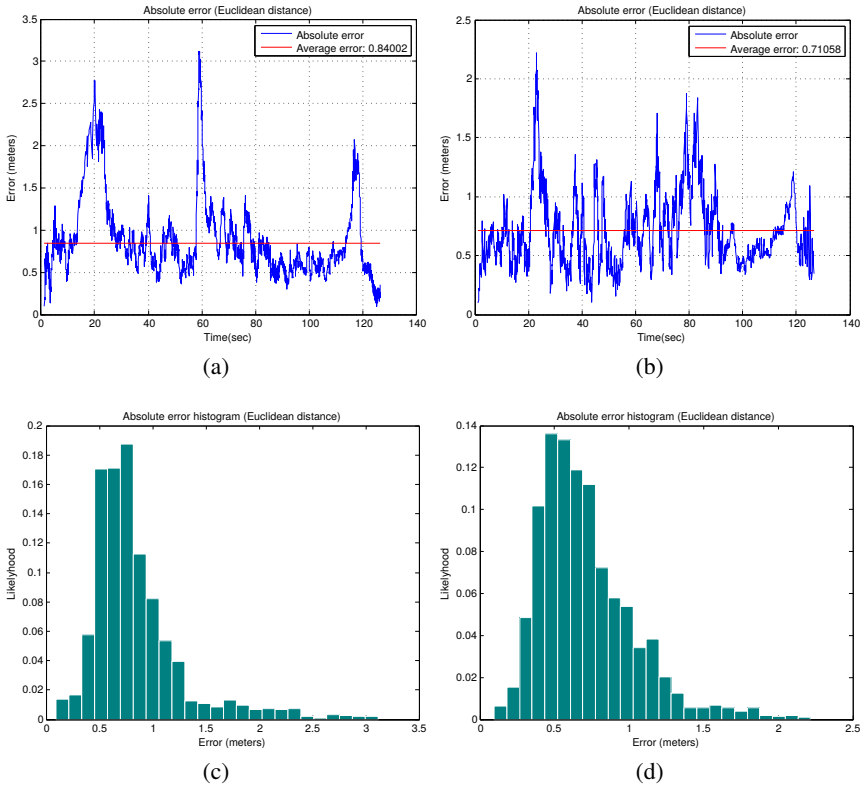


Figure 4.4: Localization results of the RO-SLAM algorithm on FADA-CATEC indoor testbed: (a) shows the absolute error when not using the pre-filtering algorithm whereas (b) shows the absolute error when using the pre-filtering algorithm. (c) shows the histogram error when not using the pre-filtering algorithm whereas (d) shows the error histogram when using the pre-filtering algorithm.

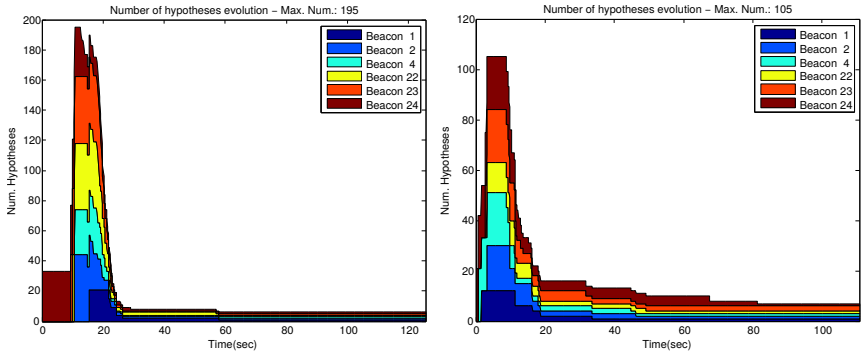
the Z axis with respect to the X and Y axis because of the UAV trajectory.

Figure 4.5 shows how the introduction of a pre-filtering algorithm increases the convergence time of the beacons due to a reduction on the number of observations given to the RO-SLAM algorithm (i.e. EKF filter).

The evolution of the hypotheses weight for one of the beacons is presented in Figure 4.6. It can be seen how almost all hypotheses are discarded after 25 seconds and at 50 seconds the filter converges to a single solution.

Table 4.1: Averaged mapping errors for 6 beacons

	Abs. Err.(m)	Horizontal Err.(m)	Vertical Err.(m)
Without pre-filtering	0.79	0.31	0.71
With pre-filtering	0.56	0.46	0.26



(a) (b)

Figure 4.5: Mapping convergence results: (a) shows the mapping convergence when not using the pre-filtering algorithm and (b) shows the results for the same experiment but using the pre-filtering algorithm. The convergence of the algorithm is shown observing the number of hypotheses (Y axis) per time-stamp (X axis).

Outdoor validation

The pre-filtering algorithm has also been tested for outdoor range-only measurements. As expected, in this case the raw range measurements are less noisy and with fewer outliers. Fig. 4.3b shows the raw and filtered range-only measurements received from the UAV to a range-only sensor with ID 6. As the figure shows, the raw measurements (blue line) has a RMSE of 2.01 meters whereas the filtered ones (red line) has a RMSE of 1.0 meters, which implies an improvement of a 50%. In this case, the improvement is lower than in the previous experiment due to a dynamic bias introduced by some reflections with the floor.

The RO-SLAM results using the pre-filtering algorithm are shown in Fig. 4.7 for localization with a global RMSE of 1.51 meters and in Fig. 4.8 for mapping results for each beacon with an averaged RMSE of 0.6 meters compared to the 1.5 meters of error of the range-only sensor employed for outdoor environments.

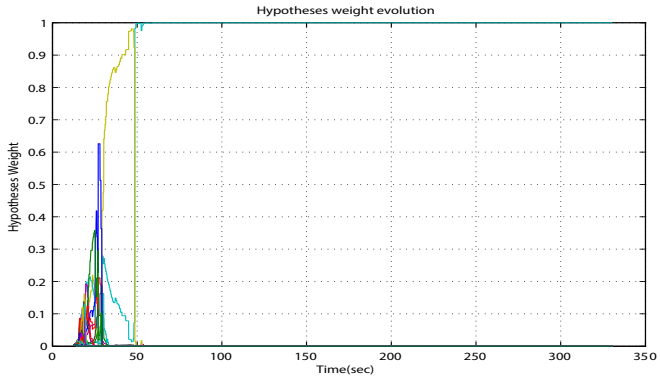


Figure 4.6: Evolution of the weights for different hypotheses of beacon 4.

4.5.2 Propagation model results

Once the results of the outlier filter are presented, this section will show how the mapping of the sensor nodes are improved by including the estimation of the measurement model. As previously introduced, the real range information from the experiments have been distorted for some nodes, using a scale factor and a small bias to corrupt the range information. The objective of this section is then analyze if the inclusion of the measurement model really improves the beacon mapping.

Figure 4.9 shows the estimated scale factor and offset for beacon 4. In this case, the real measurement had a small bias of 0.1m and a scale factor of approximately 1. It can be seen how the estimations are consistent in mean and standard deviation, having the actual solution inside the 3σ interval of the estimation.

The last part of this section aims to show the localization results obtained during the experiments described above while simultaneously the range-only sensors are being mapped together with range measurement models. The results are shown in Figure 4.10. It can be seen how the estimation follows the ground-truth with small errors most of the time. Also can be noticed the estimation is always into the 3σ interval of the estimation and the estimated deviations are well computed except for Z axis, this is because the UAV poorly trilaterates in Z.

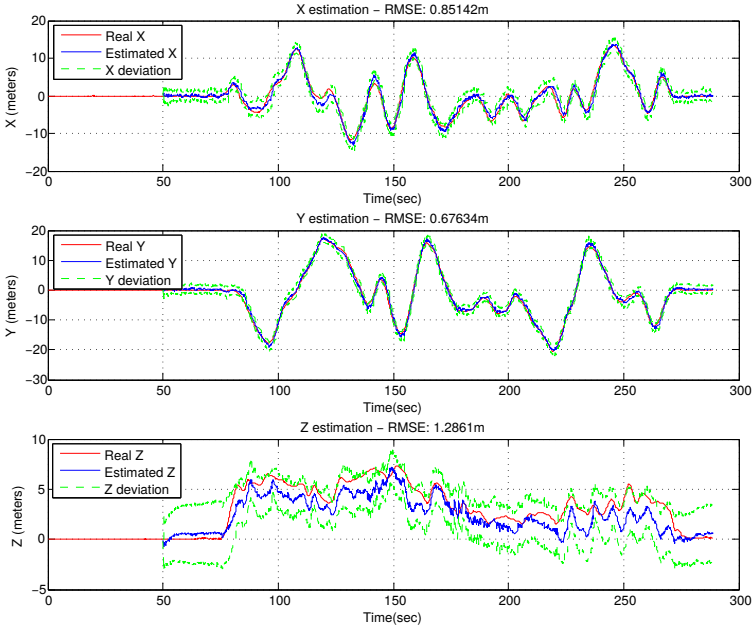


Figure 4.7: Localization results of the RO-SLAM algorithm using the proposed pre-filtering algorithm in outdoor environments.

4.6 Summary and conclusions

This chapter presented a robust method to simultaneously map the position of a set of radio range-only sensors and localize an UAV with only range measurements even in the presence of noisy measurements. The method makes use of a pre-filtering algorithm to detect and remove outliers from range-only sensors and extends the GMC model to estimate the propagation model of each radio sensor in order to correct the bias of each node. Different experiments results have been performed to validate the method. The results showed how the use of the pre-filtering method rejects several spurious range measurements which decreased the estimation error of the mapping process in approximately a 15%. Additionally, the extension of the SLAM approach presented in this chapter to model the propagation model of range measurements have demonstrated an improvement not only in the mapping results but also in the localization of the UAV.

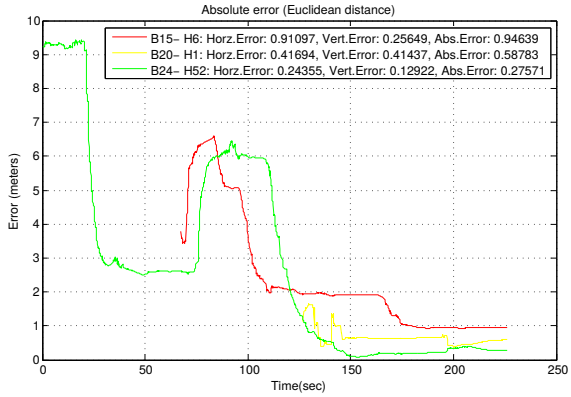


Figure 4.8: Mapping results for each beacon of the RO-SLAM algorithm using the proposed pre-filtering algorithm for outdoor environments.

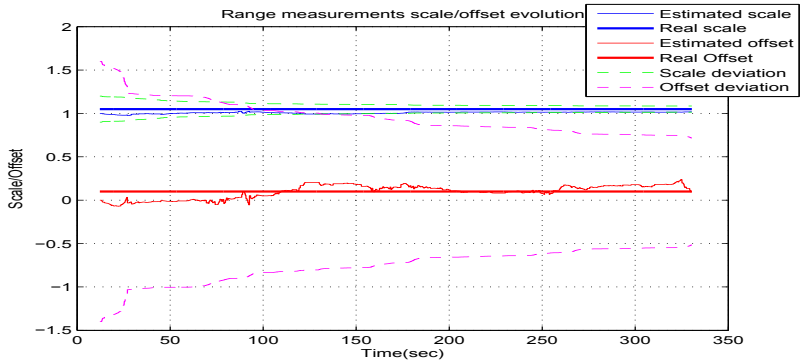


Figure 4.9: Estimated scale factor and bias for beacon 4. It can be seen how the estimations are always very close the ground-truth (straight lines) and how the actual value is into the 3σ interval of the estimation.

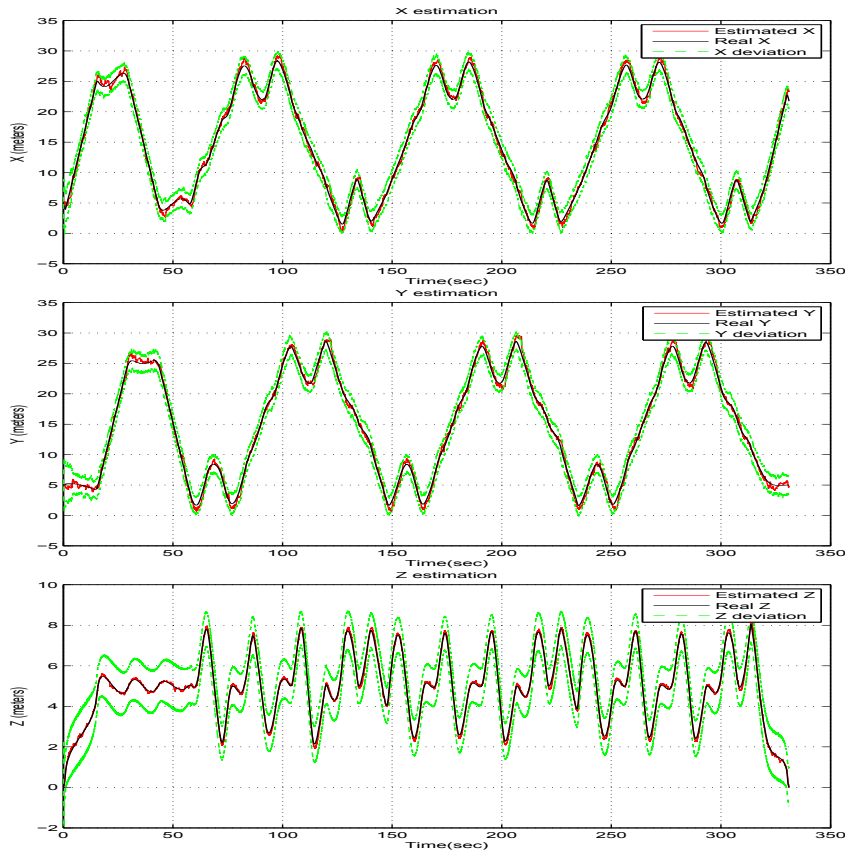


Figure 4.10: Estimated trajectory based on range-only localization and refined nodes mapping.

Chapter 5

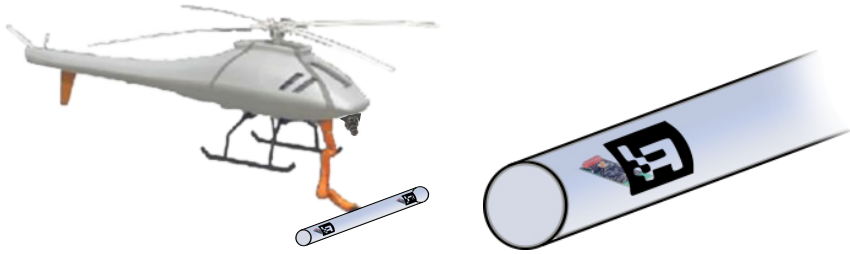
Integrating 3D RO-SLAM and visual markers

5.1 Introduction

Range sensors have been subject of research in the last decade not only in the domain of localization applications but also for SLAM systems. These devices make possible the localization of robots or other kind of objects in GPS denied environments such as indoors. They offer a low cost localization solution which does not require a direct line of sight (LOS) between each pair of sensors when employing radio systems like Wifi or Ultra Wide Band (UWB) beacons. However, the observability of range-only sensor model depends on the trajectory of the mobile robot, making visual information a perfect complement for these measurements in order to solve the ambiguities produced in those situations. In manipulation applications such as the ARCAS project, the use of range-only sensor might be used to give a rough estimation of the elements to be manipulated when their location is completely unknown and there is non line of sight (NLOS) between the camera of the vehicle and these elements, whereas visual information provides a fine estimation for manipulation tasks.

5.2 Overview and related work

In chapter 2 was presented the integration of some navigation sensors like GPS or the altimeter to refine the localization of the robot in the RO-SLAM. The mapping strategy presented in chapter 3 is suitable for aerial manipulation to offer a coarse estimation about the location of structural elements to be manipulated, so that the vehicle can approach to these structural elements and refine the position estimation employing an on-board camera and visual markers placed over the same position where radio beacons are embedded (see Figure 5.1b). The way in



(a) Aerial manipulation

(b) Embedded range sensors and visual markers

Figure 5.1: Aerial manipulation of structural elements based on embedded sensors and visual markers.

which this visual information is integrated into the filter is described in the next subsection.

As stated along this dissertation, in aerial manipulation, keeping the correlation between beacons is crucial when multiple beacons are embedded in the same structural element (see Figure 5.1). The method also improves the scalability of the system with a reduced spherical parametrization and an efficient EKF update scheme.

In general, the main drawback of 3D RO-SLAM is the observability problem associated to range-only observations which depends on the trajectory tracked by the autonomous vehicle. In [74] this issue is solved using visual information to refine the rough estimation performed by a 2D range-only mapping method which is based on a delayed EKF-SLAM. The delayed initialization algorithm of this method is based on a particle filter and the solution proposed uses a Cartesian parametrization of the beacon hypotheses which is less robust than a polar parametrization [26].

The main contributions of this chapter are the use of an optimal 3D RO-SLAM solution based on a reduced spherical parametrization integrating not only range measurements but also visual information. On the other hand, it will be proposed two different strategies to reduce the computational burden required by multi-hypotheses methods. These strategies will be compared an validated with experimental results.

The data fusion solution proposed in this chapter describes a method to include visual measurements which are based on the detection and localization of

visual markers placed on the same structural elements where the range sensors are embedded (see Figure 5.1). The method proposed is suitable for aerial manipulation tasks where an aerial robot endowed with a manipulator have to interact with several structural elements. In these cases, the system should localize the aerial robot and, at the same time, it should also map the structural elements to be manipulated. The use of visual markers allow the correction of the coarse estimation about the position of structural elements using the 3D range-only mapping algorithm detailed in the following.

5.3 Integration of visual markers

The algorithms developed in ARCAS have been used for deformable visual marker detection and localization [3]. For those markers detected, the algorithm applies the intrinsic model of the camera to get the 3D position of the marker with respect to the camera reference frame C. This relative position is integrated in the EKF by applying a transformation of this relative marker position to the frame of the vehicle V[1].

5.3.1 Correction stage

The integration of this relative positions is only computed if the filter already contains an initialized landmark with the same identification code of the marker[2]. In order to integrate this camera observation $\mathbf{v}_i^C = [x_{f_i}^C, y_{f_i}^C, z_{f_i}^C]$, it is necessary to transform the observed position of the landmark from the camera frame to the vehicle frame V as depicted in Figure 5.2. Then, the visually observed landmark position from vehicle frame $\mathbf{v}_i^V$ can be integrated in the EKF using any of the corrections methods proposed in chapter 3 (FHC, MHC or GMC).

Using correction method FHC, the observed position $\mathbf{v}_i^V$ should be compared with all hypothesis $\{\theta_{in}, \phi_{im}\}$, using Federated Information Sharing approach to split the standard deviation of the measure $\sigma_{\mathbf{v}_i}$ in $N \times M$ deviations.

[1] In the equations shown it is considered that the vehicle frame coincides with the base range-only sensor. However, including more frames to the equation would mean adding more linear transformations with static transforms. For real experimentation range-only sensors is not aligned with vehicle frame and hence this additional transformation is added to the equations.

[2] For real implementations it is common to have a mapping between markers identifier and beacons identifier.

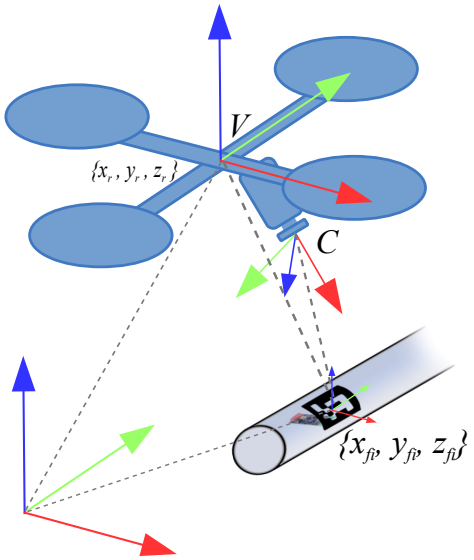


Figure 5.2: The image represents the transformation required to transform the detected visual marker position $\mathbf{v}_i^C = [x_{f_i}^C, y_{f_i}^C, z_{f_i}^C]$ from camera to vehicle frame V (i.e. position $\mathbf{v}_i^V$).

Using correction method MHC, the observed position $\mathbf{v}_i^V$ should be compared with hypotheses $\{\theta_{in}, \phi_i\}$ and $\{\theta_i, \phi_{im}\}$ (being θ_i and ϕ_i the expected values of the GMMs), and again using Federated Information Sharing approach to split the standard deviation of the measure $\sigma_{\mathbf{v}_i}$ in $N + M$ deviations.

In the case of GMC method the observation is just compared with the expected landmark position $\{\theta_{in}, \phi_i\}$. For computational reasons, GMC is the observation model implemented in this thesis[3]:

$$h(\mathbf{x}) = \begin{cases} h_i^x = x_i + \rho_i cos(\theta_i) cos(\phi_i) - x_r \\ h_i^y = y_i + \rho_i sin(\theta_i) cos(\phi_i) - y_r \\ h_i^z = z_i + \rho_i sin(\phi_i) - z_r \end{cases} \tag{5.1}$$

The Jacobian associated to this observation model would be:

[3]Temporal index t is omitted for simplification.

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & \dots & 0 & \frac{\partial h_i^x}{\mathbf{f}_i} & 0 & \dots & 0 \\ 0 & 1 & 0 & 0 & \dots & 0 & \frac{\partial h_i^y}{\mathbf{f}_i} & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 & \frac{\partial h_i^z}{\mathbf{f}_i} & 0 & \dots & 0 \end{bmatrix} \tag{5.2}$$

where the partial derivatives of $h(\mathbf{x})$ with respect to landmark parameters $\mathbf{f}_i$ would be:

$$\begin{aligned} \frac{\partial h_i^x}{\mathbf{f}_i} =& [1, 0, 0, c(\theta_i)c(\phi_i), \\ & - \omega_{\theta_{i1}} \rho_i s(\theta_i) c(\phi_i), \dots, -\omega_{\theta_{in}} \rho_i s(\theta_i) c(\phi_i), \\ & - \omega_{\phi_{i1}} \rho_i c(\theta_i) s(\phi_i), \dots, -\omega_{\phi_{im}} \rho_i c(\theta_i) s(\phi_i)] \end{aligned} \tag{5.3}$$

$$\begin{aligned} \frac{\partial h_i^y}{\mathbf{f}_i} =& [0, 1, 0, s(\theta_i)c(\phi_i), \\ & \omega_{\theta_{i1}} \rho_i c(\theta_i) c(\phi_i), \dots, \omega_{\theta_{in}} \rho_i c(\theta_i) c(\phi_i), \\ & - \omega_{\phi_{i1}} \rho_i s(\theta_i) s(\phi_i), \dots, -\omega_{\phi_{im}} \rho_i s(\theta_i) s(\phi_i)] \end{aligned} \tag{5.4}$$

$$\begin{aligned} \frac{\partial h_i^z}{\mathbf{f}_i} =& [0, 0, 1, s(\phi_i), 0, \dots, 0, \\ & \omega_{\phi_{i1}} \rho_i c(\phi_i), \dots, \omega_{\phi_{im}} \rho_i c(\phi_i)] \end{aligned} \tag{5.5}$$

5.3.2 Updating Gaussian Mixtures' weights

To update the weights of the GMMs, this chapter proposes two different approaches:

- The first algorithm prune all hypotheses of the visually detected landmark except the one with higher probability according to the visual measurement received. The observation is assumed to be very precise with respect to the hypotheses estimation of the filter which have not been pruned yet by the RO-SLAM prune strategy shown in chapter 3. The following visual measurements are used to update the most likely estimated position of the beacon according to (5.1). This algorithm fasts the convergence of landmarks at the risk of integrating a visual outlier and hence pruning bad hypotheses.

- The second algorithm updates all existing hypotheses of the beacon detected and their weights in a similar way to RO-SLAM, but now applying (5.1). Thus, the weights are updated using

$$\omega_{\theta_{in}}^{t+1} = \omega_{\theta_{in}}^{t} p(\mathbf{v}_i^{t+1}|\mathbf{x}_r^{t+1}, \mathbf{x}_i^{t+1}, \rho_i^{t+1}, \theta_{in}^{t+1}) \tag{5.6}$$

$$\omega_{\phi_{im}}^{t+1} = \omega_{\phi_{im}}^{t} p(\mathbf{v}_i^{t+1}|\mathbf{x}_r^{t+1}, \mathbf{x}_i^{t+1}, \rho_i^{t+1}, \phi_{im}^{t+1}) \tag{5.7}$$

  evaluating with TPU the probabilities $p(\mathbf{v}_i^{t+1}|\mathbf{x}_r^{t+1}, \mathbf{x}_i^{t+1}, \rho_i^{t+1}, \theta_{in}^{t+1})$ and $p(\mathbf{v}_i^{t+1}|\mathbf{x}_r^{t+1}, \mathbf{x}_i^{t+1}, \rho_i^{t+1}, \phi_{im}^{t+1})$. Other update techniques like FHU or MLU presented in Chapter 3 can also be applied, but in this thesis it was decided to use TPU due to its proved efficiency in Chapter 3. In this second approach, hypotheses are used as a mean to make the system robust against camera outliers instead of selecting the most likely which may lead to bad hypotheses due to a visual outliers.

The first method reduces considerably the computational burden of the system at the cost of a possible accuracy reduction on the estimation with the presence of visual outliers, whereas the second method is more robust against visual outliers but has a slower convergence time[4]. In practice, as it will be shown in experimental results, both methods are very similar when the aerial robot follows an optimal trajectory. In this case, it is preferred to use the first method which fasts the convergence of the map, reducing computational resources of the system.

5.4 Experimental results

For these experiments, a different data set from those presented in appendix B has been used. This outdoor experiment involves the same aerial robot, range-only sensors and camera employed for experiments detailed appendix B. In this case the dataset has been recorded in the Engineering School at the University of Seville (Spain). In these experiments some visual markers have been conceived in order to validate the approach presented in this chapter. The main objective

[4]The integration of visual markers fasts the convergence of landmarks in both cases, but second method is slower with respect first one since it updates the weights instead of keeping the most probable.

of these experiments is to demonstrate the suitability of the approach when real sensors with noisy measurements are used. The experimental results will be presented at two different levels: pure mapping experiments and full SLAM.

The mapping experiments are oriented to show how the approach behaves when the localization of the aerial robot is solved, and the problem is reduced to map the position of the range nodes in the space. This is a typical situation in indoors experiments, when very precise positioning systems such as VICON or OptiTrack are used, or when RTK positioning systems are used outdoors.

SLAM experiments will consider the localization of both the aerial vehicle and the range nodes without the integration of precise localization systems. The section will show results based only on local sensors such as the baromenter, range-only sensors and visual markers, and results also integrating standard GPS in single configuration.

Prior to the discussion of the experimental results, next section describes the setup of the experiment describing all the elements involved: aerial robot, sensors, ground-truth, etc.

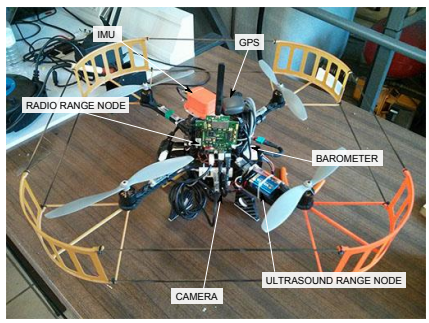
5.4.1 Experimental setup

The complete experimental setup is composed by the following elements:

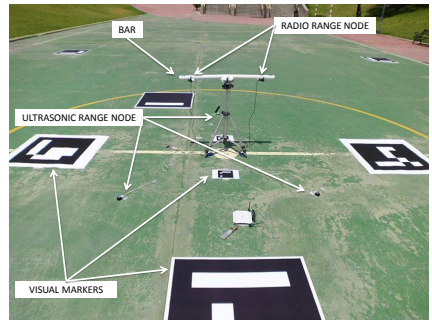
- An aerial robot with the following on-board sensors (see Figure 5.3b): GPS in single configuration, a radio based range sensor, an ultrasound based range sensor, a barometer and a visual camera pointing downwards.
- A bar with two range sensors based on radio and two small visual markers (see Figure 5.3c).
- Five range sensors based on radio. Three anchors and two beacons mounted in the bar (see Figure 5.3c).
- Three range-only sensors based on ultrasound placed at known positions in the scenario (see Figure 5.3c).
- Several visual markers with different size deployed all over the scenario at known positions (see Figure 5.3a).

In this case, the only positioning information fused with range-only observations was the single GPS on-board. In order to provide a good ground-truth for

(a)



(b)



(c)

Figure 5.3: Experimental setup: (a) The scenario of the outdoor experiment with visual markers on the floor for ground truth estimation. (b) Sensors on-board the aerial robot. The system is equipped with GPS in single configuration, a range sensor based on radio, a range sensor based on ultrasounds, a visual camera looking downwards, an inertial measurement unit and a barometer. (c) Experiments area with several elements deployed: ultrasound and radio range sensors, visual markers at known positions and a bar whose position will be estimated based on range sensors and visual markers.

the position of the aerial robot, the real position of all the visual markers placed on the floor were registered and the position of the vehicle with respect to them computed as a RANSAC PnP problem using Levenberg-Marquardt optimization. Thus, the absence of vehicle ground truth in result figures is due to a lack of markers visibility from the vehicle camera. This setup allows errors in the order of 10 cm when the markers occupy the 70% of the image or when several markers are detected in the same image.

During the experiments, the aerial robot was flying around and close to the bar in order to detect the visual markers on top of it. Next sections will explain the mapping and SLAM results obtained with Matlab. A video summarizing the experiments and the results obtained with an implementation on ROS framework can be downloaded from http://grvc.us.es/staff/felramfab/icra2014/video.mp4.

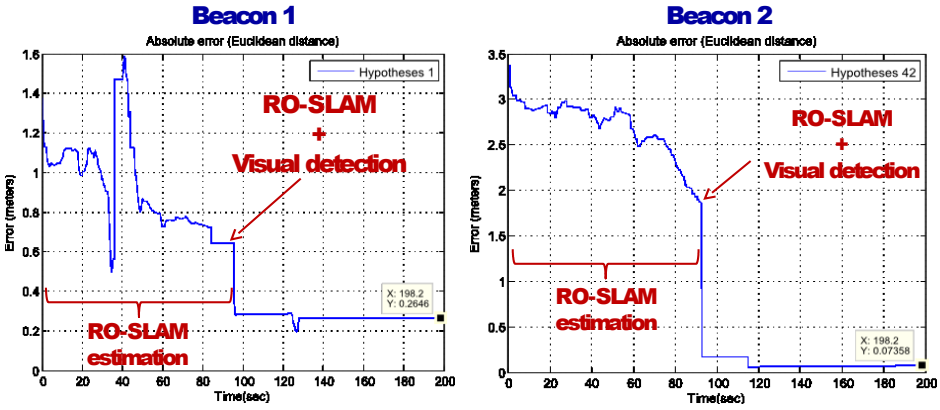


Figure 5.4: Mapping results with visual markers. Error evolution of the best hypothesis selected by RO-SLAM algorithm for the two beacons attached to the bar before and after introducing visual marker observations.

5.4.2 Mapping with visual markers results

The goal of this experiment was to estimate the position of the range sensors attached to the bar assuming that the position of the vehicle is known. For this purpose, the robot localization based on visual markers on the floor is used as robot true position, and the range nodes and visual markers on the bar are used to estimate their position. This experiment allows showing the convergence and proper behavior of the approach when the robot position is known, which is very usual in indoor setups or outdoors with accurate position systems such as the RTK position system.

Figure 5.4 presents the error evolution of the best hypothesis selected by RO-SLAM algorithm for both range sensors attached to the bar. All the hypotheses are initialized around the robot position with the first range measurement. After some trilateration, most position hypotheses have been deleted for beacon 1, while for beacon 2 have already converged to a single solution. Figure 5.4 shows the results right after the visual marker information attached to each node has been integrated using the prune hypotheses method. It can be seen how both positions suffer a huge improvement on their estimation since the very first visual marker detection. After integrating more range and visual information the absolute errors are about 7 cm for beacon 2 and 26 cm for beacon 1.

It is important to mention that the minimum distance from the aerial robot to

the marker was 1.5 m during the mapping experiments for safety. Visual markers detection on the marker's bar could be improved if the aerial robot were closer, and this improvement can be directly translated to the mapping process.

Both methods for visual marker detection have been implemented (prune hypotheses and weighting), but they yield to the same results with slightly differences, so the weighting results are not shown in the chapter. In general, results will be very similar because the hypotheses quickly converge to a single one thanks to the good precision of the visual markers measurements. However, the weighting method will be more robust if the visual marker measurement is subject to outliers, which is not usual in the ARCAS method proposed for marker detection. Otherwise the prune hypotheses method is the best option from the computational point of view.

5.4.3 SLAM with visual markers results

This section presents the results when the SLAM filter is used. The method integrate range only measurement from two different type of sensors at known positions and three radio based sensors with standard deviation of 1.5 m approximately. The SLAM filter also integrates barometric information in order to estimate the altitude. The positions of two radio nodes attached to the bar are also mapped and the position of the visual markers placed on the bar when they are detected are also integrated.

Figure 5.5 shows the localization results of the SLAM filter. The global root mean square (RMS) error is about 1.7 m, but the error is below this threshold most of the time as it can be seen in the figure. The ground-truth showed in the figure is the estimation provided by the visual marker detector. It can be seen how the estimation follows the real position of the vehicle most of the time with small errors. It is remarkable that no GPS have been used on the robot pose estimation. The localization of the nodes on the bar converges to single hypotheses thanks to the visual marker integration and they are localized with errors of 0.7 and 0.6 m respectively.

Figure 5.6 shows the results of the same experiment, but integrating a GPS in single configuration with the usual 2.5 m circular error probability (CEP). It can be seen how the RMS error is almost reduced to the half, 1.06 m. The mapping of nodes on the bar is slightly improved in one of them with localization errors of 0.7 and 0.4 m respectively. In general, the SLAM filter behaves better because the prediction model does not constraint enough the space of possible solutions,

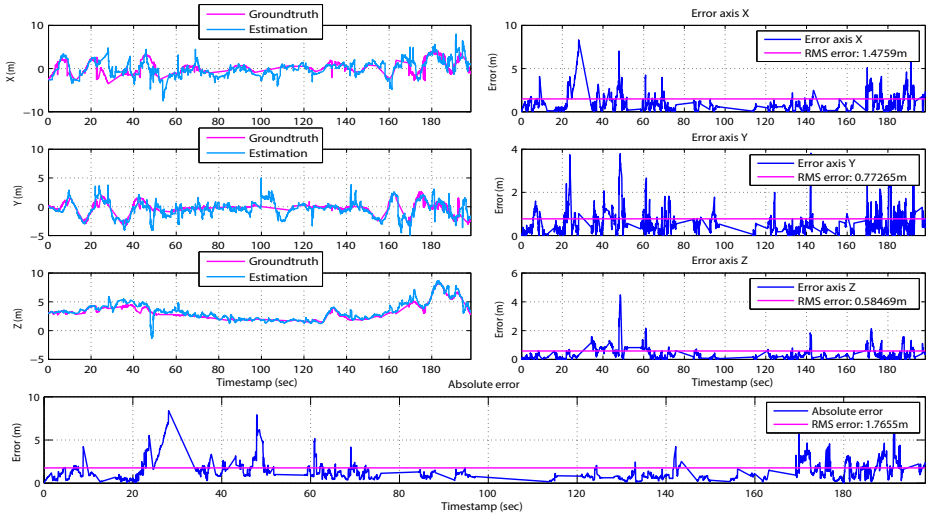


Figure 5.5: SLAM results without GPS. Estimated position of the aerial robot with the SLAM filter. The position integrates range measurements from three radio and three ultrasonic devices and a barometer for altitude estimation. No GPS is used for position estimation. The ground-truth is computed based on the detection of visual markers placed in the floor at known locations. The graphs show the estimated X, Y and Z values together with the ground truth. The RMS error per axis and the global error are also shown.

because of the random walk predictive model used in this case. If other predictions are used, as visual odometry, the results will not differ significantly with respect to the SLAM without GPS. It is important to remark that from seconds 100 to 120 there was no ground-truth because there were no markers in the field of view of the camera on-board the aerial robot. The mean value has been plotted in order to have an estimation.

5.5 Summary and conclusions

This chapter detailed the data fusion technique used for the RO-SLAM approach presented in chapter 3 with different navigation sensors for the localization of the robot and putting especial attention on visual markers integration for the map refinement. In this chapter, RO-SLAM was used as a coarse estimation of landmarks which is latter refined with the integration landmarks position

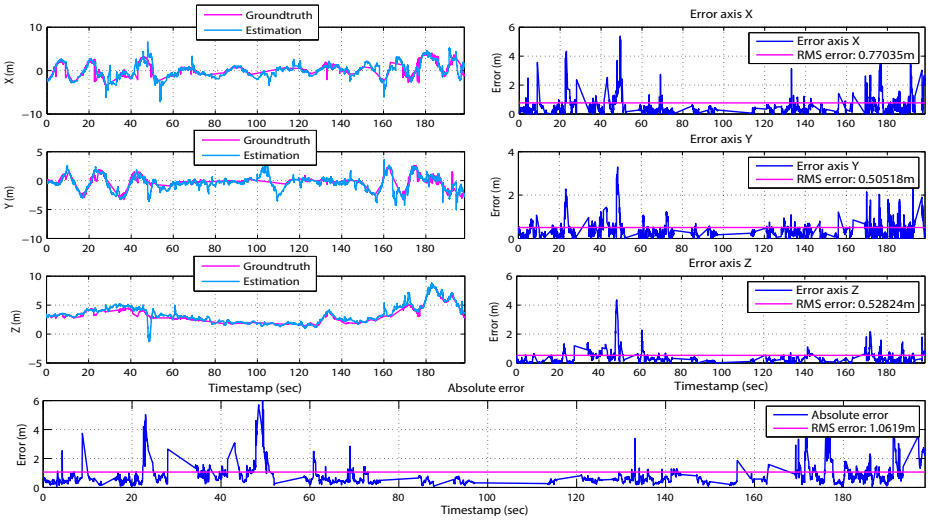


Figure 5.6: SLAM results with GPS. Estimated position of the aerial robot with the SLAM filter. The position integrates range measurements from three radio and three ultrasonic devices, an IMU, a barometer for altitude estimation and a GPS in single configuration. The ground-truth is computed based on the detection of visual markers placed in the floor at known locations. The graphs show the estimated X, Y and Z values together with the ground truth. The RMS error per axis and the global error are also shown.

information coming from a visual marker detector. This chapter was based on the ARCAS project application, where radio range-only sensors are integrated in bars and visual markers are placed on top of this range-only sensors to fuse range-only sensors with visual markers observation.

The marker detector used during the experiments is based on a deformable markers. This is necessary in the case of ARCAS project since markers are placed on cylindrical bars and hence the markers detector must be able to cope with this kind of deformations. The detector is supposed to return the relative position of the landmars with respect the camera frame, which is later transformed to the vehicle frame before applying the correction stage of the filter. The chapter introduced how use FHC and MHC correction schemes detailed in chapter 3 but only GMC method is selected for implementation due to the efficiency of the method.

For the update of weights, two algorithms were proposed. The first algorithm

prunes all hypotheses of the landmark except the one which gives a higher probability with the visual observation. The second method applies the TPU update strategy presented in chapter 3. The first method allows to reduce the computational requirements of the system by pruning all hypotheses at the risk of integrating a visual outlier which might select the wrong landmark hypotheses. The second method is more robust against outliers but takes more time to converge the landmarks' estimation.

Experimental results showed a reduced error on mapping results. The results also showed how the localization error can be decreased by integrating navigation sensors for outdoor environments like GPS or other sensors for indoor and outdoor environments like a barometric altimeter or acoustic range-only sensors.

Chapter 6

Cooperative 3D RO-SLAM

6.1 Introduction

This chapter proposes a decentralized algorithm for range-only SLAM where aerial vehicles share their local belief to reduce the uncertainty on the map estimation. The method is based on a easy-to-implement and efficient methodology which tries to find the intersection between two aerial vehicles estimation to prune those hypotheses which does not intersect in this shared estimation as shown in Figure 6.1. In this chapter, two methods are proposed, comparing them in simulated and real experiments involving multiple aerial vehicles and several range-only sensors. Furthermore, the method is specially designed to work with the reduced parametrization presented in chapter 3.

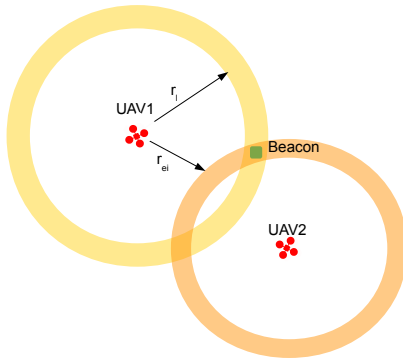


Figure 6.1: Intersection of two aerial vehicles map estimation for a single landmark (range-only beacon). The intersection of both estimations defines the most probable places of the landmark (beacon) and makes possible the reduction of the number of hypotheses in both aerial vehicles, easing the convergence of the filter and reducing the computational burden of the method.

6.2 Overview and related work

To tackle with the flip ambiguity problem, a possible solution consist on sharing the map belief of each aerial vehicle with others, so that each individual robot can reduce the uncertainty on its local estimation of landmarks (range-only sensors) by fusing it with other robot estimations. In that sense, some authors propose the use of decentralized estimation strategies [28] in order to propagate individual sensors or robot beliefs to others using a Belief Propagation (BP) algorithms in order to compute marginals distribution without using a centralized approach. Other approaches are based on the fusion of local estimations of a landmark. However, when sharing information between aerial vehicles it is desirable to know the cross correlation existing between both estimations, when this information is not known, in order to avoid some inconsistencies which might make the filter overconfident, some authors propose the use of the Covariance Intersection (CI) [18, 63] method which gives a close approximation to the optimal solution as compared to the one obtained when cross correlations are known. However, this method is not optimal for centralized EKF-SLAM filters, specially in multi-hypotheses approaches. This chapter proposes a more easy-to-implement and efficient algorithm which takes advantage of the reduced parametrization proposed in chapter 3.

6.3 Multi-hypotheses map fusion

The method proposed in this chapter uses a EKF-SLAM framework where each robot estimates the localization and map estimation in a local state vector. Thus, the state vector $\mathbf{x}$ of each aerial robot is composed by its estimated position $\mathbf{x}_r = [x_r, y_r, z_r]^T$ and the position of m landmarks $\mathbf{f}_i$ (range-only sensors) which are in its field of view:

$$\mathbf{x} = [\mathbf{x}_r, \mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_m]^T \tag{6.1}$$

Taking into account the reduced parametrization described in chapter 3, this section proposes two methods for decentralized map fusion which makes possible to share the aerial vehicles local estimations to reduce the local uncertainty and without introducing filter inconsistencies. Both methods are summarized in Algorithm 4 and Algorithm 5. Given two maps estimations m_e and m_l from external robot e and the local robot l respectively, these algorithms reduces the number of hypotheses in m_l by fusing the belief on those landmarks which

appears in both maps. The output of the algorithm is then the fused map m'_l. In this algorithm, those local landmarks which do not appear in external map keep they estimations as they were in m_l.

6.3.1 Method 1: Intersection pruning

In this method, the first part of the algorithm transforms the spherical coordinates of each external hypotheses $\mathbf{h}_i^e$ of beacon $\mathbf{b}_i$ so that they get relative to the central point $\mathbf{x}_i^l$ of landmark parameters $\mathbf{f}_i^l$ instead of the original center $\mathbf{x}_i^e$ of landmark parameters $\mathbf{f}_i^e$, i.e. transform the coordinate system of h_i^e so that the center of the spherical representation is the same of local landmark $\mathbf{f}_i^l$. With this change of central point, each hypotheses of beacon $\mathbf{b}_i$ in map m_e have a different radius $r_{h_i^e}$ per hypotheses $\mathbf{h}_i^e$. Then, as the radius to each external hypotheses r_e^i might be different to the local hypotheses r_l^i, the algorithm compares the radius of each external transformed hypotheses with the local radius. If the difference between both radius is below a certain threshold r_{th}, then the hypotheses are considered to intersect with the local landmark belief (in practice this threshold gives good results for a value equal to standard deviation of range-only measurements $r_{th} = \sigma_r$). By comparing both radius, the algorithm keeps only those local hypotheses which lies inside the intersection between the local and external beliefs.

Later, for each external hypotheses which intersect with the local belief, the algorithm finds the most similar local azimuth θ_l^i and elevation ϕ_l^i samples with respect the external azimuth θ_e^i and elevation ϕ_e^i samples. The matching procedure is performed taking the local sample which gives the biggest Bhattacharyya coefficient $\mathcal{BC}(p, q)$ with respect the external samples. Thus, the method takes into account not only the Euclidean distance between samples but also their associated variances. In this case, as azimuth and elevation samples follow a Gaussian distribution $\mathcal{N}(\mu, \sigma)$, the local best match Θ_l^i and Φ_l^i are computed as:

$$\delta_i^l = \arg\max_{\delta_l^i} \mathcal{BC}(\delta_i^l, \delta_e^i) \tag{6.2}$$

where the expression is equal for $\delta_i^l = \theta_i^l$ or $\delta_i^l = \phi_i^l$.

The Bhattacharyya coefficient $\mathcal{BC}$ measures the amount of overlap between two statistical samples. This coefficient is calculated for samples δ_l^i and δ_i^e as:

$$\mathcal{BC}(\delta_l^i, \delta_e^i) = \sqrt{\frac{\sigma_{\delta_l^i}\sigma_{\delta_e^i}}{\sigma_\delta}} e^{-\frac{1}{8}\frac{(\delta_l^i - \delta_e^i)^2}{\sigma_\delta}} \tag{6.3}$$

Algorithm 4: Decentralized map fusion: Method 1

Data: m_e, m_l
Result: The fused map m'_l

begin
  $m'_l \longleftarrow \varnothing$;
  for *each beacon* $\mathbf{b}_i$ *in* m_l *but not in* m_e **do**
    $m'_l \longleftarrow b_i$
  for *each beacon* $\mathbf{b}_i$ *in* m_e *and* m_l **do**
    // Set of new local azimuth and elevation angles
    $T_l \longleftarrow \varnothing$;
    $P_l \longleftarrow \varnothing$;
    $r_i^l \longleftarrow$ Radius of $\mathbf{b}_i$ in m_l;
    for *each hypothesis* $\mathbf{h}_i^e$ *of* $\mathbf{b}_i$ *in* m_e **do**
      $\mathbf{sph}_i^e \longleftarrow \mathbf{h}_i^e$ centered at $\mathbf{x}_i^l$;
      $r_i^e \longleftarrow$ Radius of $\mathbf{sph}_i^e$;
      $diff \longleftarrow$ abs(r_i^l - r_i^e);
      // Check intersection of local and external hypotheses
      if $diff \leq r_{th}$ **then**
        $\Theta_i^l = \arg\max_{\theta_l^i} \mathcal{BC}(\theta_i^l, \theta_e^i)$;
        $T_l \longleftarrow \mathcal{N}(\Theta_i^l, \sigma_{\Theta_i^l})$;
        $\Phi_i^l = \arg\max_{\phi_l^i} \mathcal{BC}(\phi_i^l, \phi_e^i)$;
        $P_l \longleftarrow \mathcal{N}(\Phi_i^l, \sigma_{\Phi_i^l})$;
    // Keep only local hypotheses which matched
    $\Omega_{T_l} \longleftarrow$ weights of modes in T_l;
    $\Omega_{P_l} \longleftarrow$ weights of modes in P_l;
    // Normalize new weights
    $\Omega_{T_l} \longleftarrow Normalize(\Omega_{T_l})$;
    $\Omega_{P_l} \longleftarrow Normalize(\Omega_{P_l})$;
    // Update belief of landmark $\mathbf{b}_i$
    $m'_l \longleftarrow \{\mathbf{x}_i^l, r_i^l, T_l, P_l, \Omega_{T_l}, \Omega_{P_l}\}$;

where

$$\sigma_\delta = \frac{1}{2}\sigma_{\delta_l^i} + \frac{1}{2}\sigma_{\delta_e^i} \tag{6.4}$$

After computing all matches between external and local samples, the algorithm prunes all local samples which do not match with any external sample and updates the weights of matched samples normalizing them. By pruning these local samples, the system converges faster to a single hypotheses. Additionally, the algorithm is able to solve the flip ambiguity of the RO-SLAM approach by intersecting the beliefs of two robots with different trajectories.

6.3.2 Method 2: Map fusion

This method extends the first method by including in the set of local matched hypotheses m_l' those external θ_i^e and ϕ_i^e samples which made matching with local ones. First of all, the weights of external samples need to be normalized. Later, this chapter proposes to weight local estimations over external ones by a factor of α (typically 0.6 in practice [18]) in a similar way as it is done in Covariance Intersection methods in order to cancel the effect of double counting information or *rumor propagation*:

$$\begin{aligned} \omega_{\theta_i^l} &= \alpha\omega_{\theta_i^l} & \omega_{\phi_i^l} &= \alpha\omega_{\phi_i^l} \\ \omega_{\theta_i^e} &= (1-\alpha)\omega_{\theta_i^e} & \omega_{\phi_i^e} &= (1-\alpha)\omega_{\phi_i^e} \end{aligned} \tag{6.5}$$

The introduction of new samples in the local filter may help each aerial robot to have better re-sampling of the landmark probability distribution by sharing hypotheses with other aerial vehicles. This method might lead the filter to converge slowly if the fusion is performed very frequently since new samples are going to be included for each fusion attempt. Then, in practice it is better to use this fusion method only when the external map has converged to a certain number of hypotheses. On the other hand, the use of the merge strategy presented in chapter 3 for similar Gaussian modes will fuse those local and external hypotheses which are very similar avoiding duplicated hypotheses in the filter but preserving the probability distribution.

At the end of both methods the algorithm normalize the new local azimuth and elevation samples included in m_l'. Experimental results performed at laboratory showed that in general *Method 2* gives better results than *Method 1* in mapping

Algorithm 5: Decentralized map fusion: Method 2

Data: m_e, m_l
Result: The fused map m'_l

begin
  $m'_l \longleftarrow \varnothing$;
  for *each beacon* $\mathbf{b}_i$ *in* m_l *but not in* m_e **do**
    $m'_l \longleftarrow b_i$
  for *each beacon* $\mathbf{b}_i$ *in* m_e *and* m_l **do**
    // Set of new local azimuth and elevation samples
    $T_l \longleftarrow \varnothing$;
    $P_l \longleftarrow \varnothing$;
    // Set of intersected external azimuth and elevation samples
    $T_e \longleftarrow \varnothing$;
    $P_e \longleftarrow \varnothing$;
    $r_i^l \longleftarrow$ Radius of $\mathbf{b}_i$ in m_l;
    for *each hypothesis* $\mathbf{h}_i^e$ *of* $\mathbf{b}^i$ *in* m_l *and* m_e **do**
      $\mathbf{sph}_i^e \longleftarrow \mathbf{h}_i^e$ centered at $\mathbf{x}_i^l$;
      $r_i^e \longleftarrow$ Radius of $\mathbf{sph}_i^e$;
      $diff \longleftarrow$ abs(r_i^l - r_i^e);
      // Check intersection of local and external hypotheses
      if $diff \leq r_{th}$ **then**
        $\{T_e, P_l\} \longleftarrow \{\theta_i^e, \phi_i^e\}$ from sph_i^e;
        $\{\Omega_{T_e}, \Omega_{P_e}\} \longleftarrow \{\omega_{\theta_e^i}, \omega_{\phi_e^i}\}$;
        // Find best match between local and external samples
        $\Theta_i^l = \arg\max_{\theta_l^i} \mathcal{BC}(\theta_i^l, \theta_e^i)$;
        $\Phi_i^l = \arg\max_{\phi_l^i} \mathcal{BC}(\phi_i^l, \phi_e^i)$;
        $\{T_l, P_l\} \longleftarrow \{\mathcal{N}(\Theta_i^l, \sigma_{\Theta_i^l}), \mathcal{N}(\Phi_i^l, \sigma_{\Phi_i^l})\}$;
    // Add matched external samples
    $\{T_l, P_l\} \longleftarrow \{T_l \bigcup T_e, P_l \bigcup P_e\}$;
    // Ponderate local samples over external samples
    $\{\Omega_{T_l}, \Omega_{P_l}\} \longleftarrow$ weights of modes in T_l and P_l;
    $\{\Omega_{T_e}, \Omega_{P_e}\} \longleftarrow \{Normalize(\Omega_{T_e}), Normalize(\Omega_{P_e})\}$;
    $\Omega_{T_l} \longleftarrow \alpha\Omega_{T_l} \bigcup (1-\alpha)\Omega_{T_e}$;
    $\Omega_{P_l} \longleftarrow \alpha\Omega_{P_l} \bigcup (1-\alpha)\Omega_{P_e}$;
    // Normalize all weights
    $\{\Omega_{T_l}, \Omega_{P_l}\} \longleftarrow \{Normalize(\Omega_{T_l}), Normalize(\Omega_{P_l})\}$;
    // Update local belief of beacon $\mathbf{b}_i$
    $m'_l \longleftarrow \{\mathbf{x}_i^l, r_i^l, T_l, P_l, \Omega_{T_l}, \Omega_{P_l}\}$;

due to the re-sampling process around the real position which tends to get closer to the real landmark position. However, for localization, the results are virtually the same between both methods.

The following experiment will show the results of *Method 2* since, as explained above, this method tends to be more accurate and, in the end, it is an extension of *Method 1* which incorporates matched external samples into the local filter.

6.4 Simulated and real experiments

Simulation experiments

These simulation experiments have been conceived in order to analyze the behavior of the proposed decentralized RO-SLAM approach. For this purpose, four aerial vehicles have been simulated using Gazebo under the Robot Operating System (ROS) framework. Range-only measurements are also simulated with a range-only simulator developed in this thesis for a set of sensors deployed at fixed positions in the simulation arena.

The ground-truth position of the aerial vehicles and the sensors deployed in the environment is presented in Figure 6.2. It can be seen how the trajectories are selected in order to avoid possible vehicle collisions. Red squares in Figure 6.2 stand for the position of four range-only sensors used as anchors (sensors with known position). The real position of the range-only sensors to be estimated are marked as blue squares in the figure.

Figure 6.3 shows the estimated position error for UAV1 and UAV2 extracted from the RO-SLAM when no decentralized fusion is included into the estimation. This error is computed as the absolute distance between the estimated UAV position and the ground-truth. It can be seen how the average errors are 0.7, 0.5, 0.4 and 0.3 meters respectively, and how the errors are reduced as the position of the different beacons are refined into the RO-SLAM filter in each aerial vehicle.

On the other hand, Figure 6.4 shows the estimated error for all UAVs when the decentralized fusion proposed in this chapter is used. This means the vehicles exchange landmark information during their trajectories, fusing each other estimation. It can be seen how the average error in the aerial vehicles localization is smaller than the results presented in Figure 6.3. The errors with decentralized fusion are almost the half of the errors in single configuration. In addition, it can be seen how the errors are stable and only experiment a small peak at the beginning

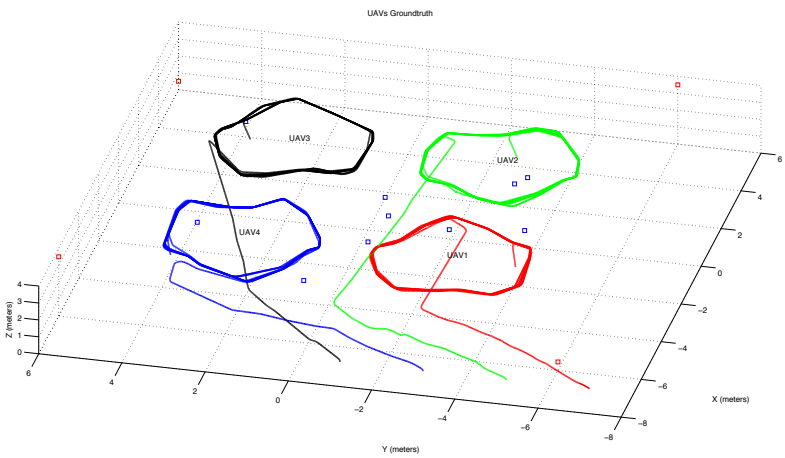


Figure 6.2: Ground-truth used for simulation experiments. It can be seen the trajectory of the four aerial vehicles used in the experiment. Blue squares stand for range-only sensors which position is estimated. Red squares stand for range-only anchors with known positions.

of the estimation due to the uncertainties introduced by the unknown position of the landmarks, but landmarks are quickly estimated and used to improve the robot localization.

The improvements in the convergence to a single hypothesis for all the estimated landmarks can be seen in Figure 6.5. When the RO-SLAM includes decentralized information from the other aerial vehicles, the landmarks are better trilaterated, so that the estimation is improved. It can be seen how the decentralized approach dramatically reduces the number of hypotheses thanks to the fusion with other robot's information. Notice the peaks at seconds 25 and 140. They are produced by the discovery of landmarks 2 and 4 at these moments, producing an increment in the number of hypotheses.

Real Experiments

The approach has also been validated using data from real experiments involving a set of range-only sensors deployed in the environment and two aerial vehicles equipped with range sensors. The experiment setup is the usual VICON setup at CATEC testbed. The aerial vehicles flew in the indoor testbed of FADA-CATEC

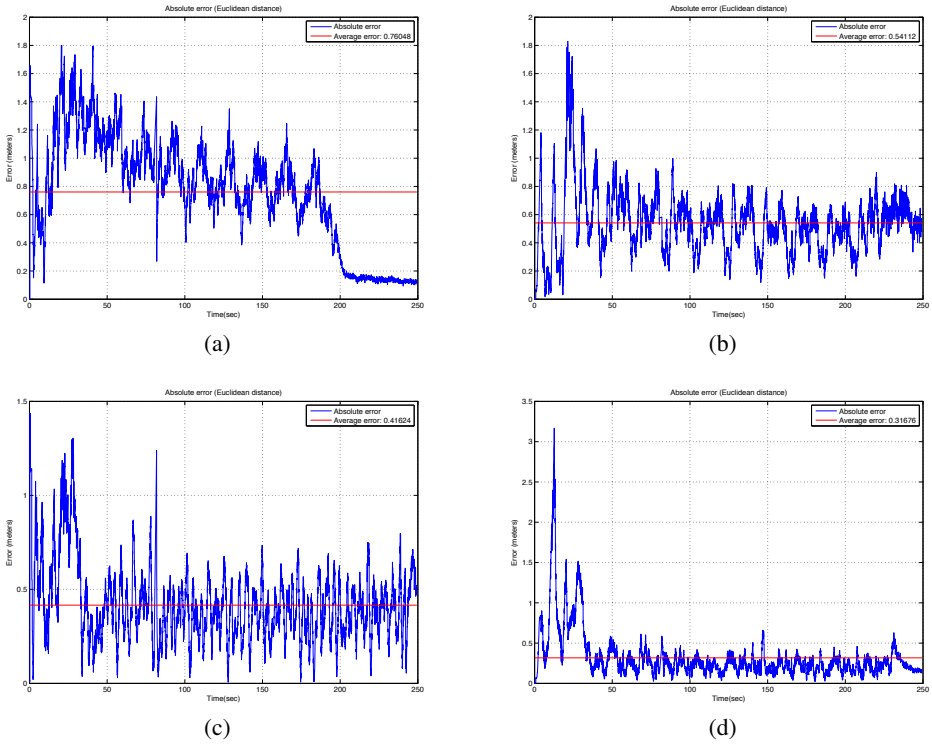


Figure 6.3: Estimated absolute 3D error of the aerial vehicles localization in simulation when no fusion of information is performed into the RO-SLAM filter. The blue solid line shows the instant error and the red solid line represents the average error. (a) UAV1, (b) UAV2, (c) UAV3 and (d) UAV4

(Seville, Spain) and the ground-truth position of the vehicles were computed using a visual tracking system (VICON). The range-only sensors were also localized using the VICON system and their position recorded in order to benchmark the position estimation of the multi-SLAM approach presented.

The implementation of methods is programmed in C++ under the Robot Operating System (ROS) framework[1]. The range-only sensors used in the experiments are manufactured by Nanotron [77]. They are low-cost sensors based on

[1] A video with the real experiments and algorithm implementation in ROS can bee downloaded from http://grvc.us.es/staff/felramfab/icra2015/video.mp4

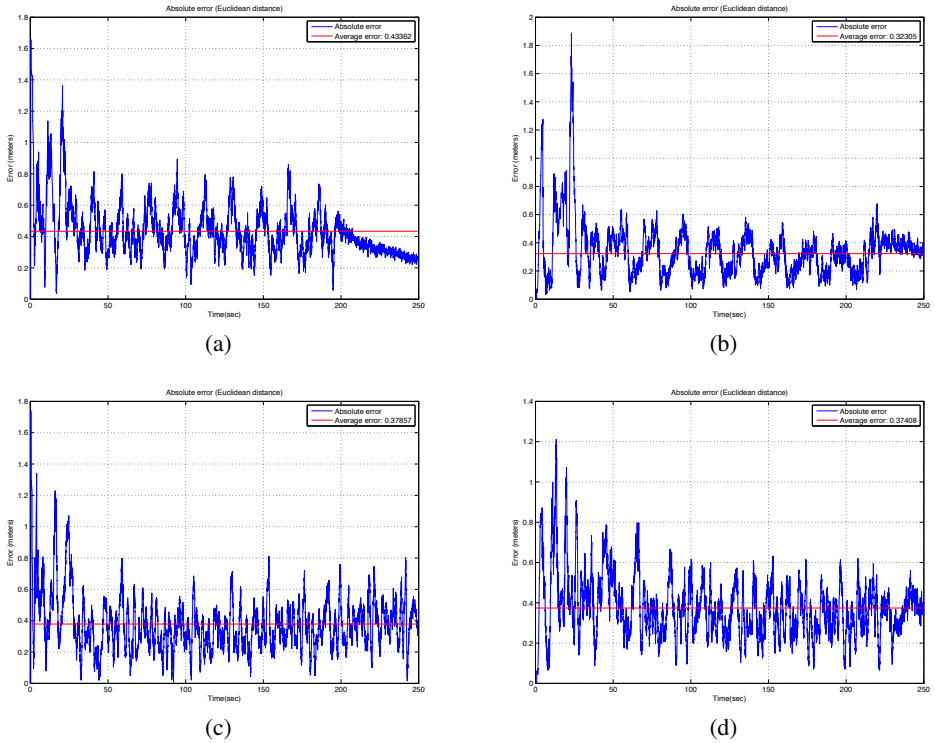


Figure 6.4: Estimated absolute 3D error of the aerial vehicles localization in simulation when the approach for decentralized RO-SLAM presented in this chapter is used. The blue solid line shows the instant error and the red solid line represents the average error. (a) UAV1, (b) UAV2, (c) UAV3 and (d) UAV4

radio time-of-flight with measurement standard deviation of $1m$ approximately. Range measurements were limited to 30 meters as the number of outliers increase linearly with the distance between the sensors. A pre-filtering algorithm have been used in all experiments for range observations using the algorithm described in [37].

The actual UAVs trajectories together with beacon and anchor positions are shown in Figure 6.6. For these experiments 6 anchors have been used due to the large errors introduced by the range-only sensors measurements (standard deviation of $1m$) and also to the lack of odometry into our UAVs. Thus, Figure

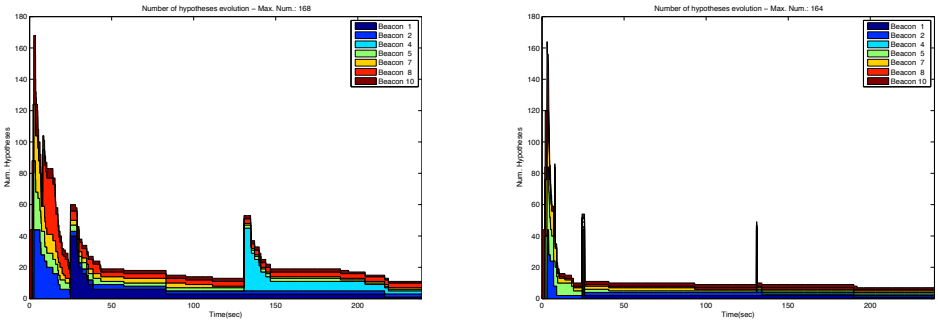


Figure 6.5: Evolution of the total number of hypotheses for the landmarks in UAV1 for the case of single configuration (left) and decentralized fusion (right).

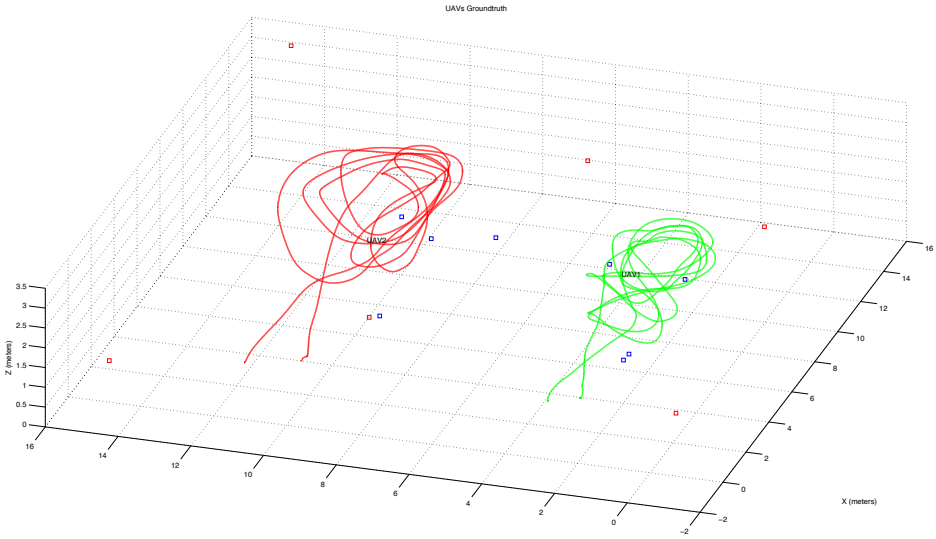


Figure 6.6: Ground-truth of the real experiments carried out at FADA-CATEC indoor testbed. It can be seen the trajectory of the two UAVs used in the experiment. Blue squares stand for range-only sensors which position is estimated. Red squares stand for range-only anchors with known positions.

6.7 and Figure 6.8 show the errors of the estimated trajectory with respect the ground-truth. It can be seen how the error is slightly smaller (about 15%) in the decentralized estimation than in the single configuration.

Figure 6.9 shows the estimated error in the UAV1 landmark localization for

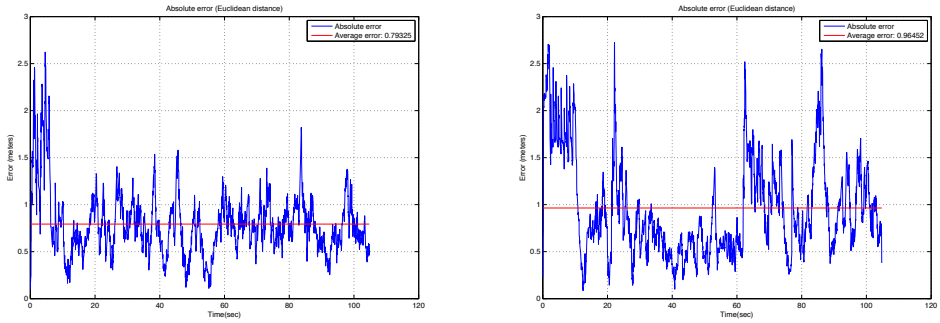


Figure 6.7: Estimated absolute 3D error of the aerial vehicles localization in real experiments when no fusion of information is performed into the RO-SLAM filter. The blue solid line shows the instant error and the red solid line represents the average error. (Left) UAV1. (Right) UAV2

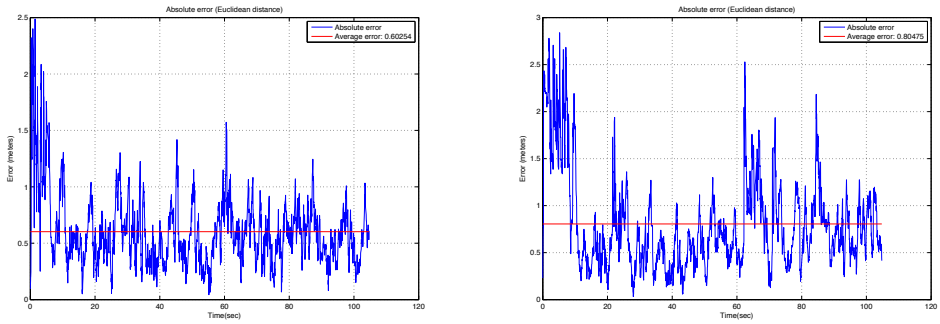


Figure 6.8: Estimated absolute 3D error of the aerial vehicles localization in real experiments when the approach for decentralized RO-SLAM presented in this chapter is used. The blue solid line shows the instant error and the red solid line represents the average error. (Left) UAV1. (Right) UAV2

both, single and decentralized approaches. In both cases the localization error is large until the filter converges to a single solution. Nevertheless, the landmark localization error is in the order of the range-only sensor standard deviation of approximately $1m$. Notice how landmark 1 do not converge to the right solution in the case of single estimation, introducing errors of $2.5m$ approximately.

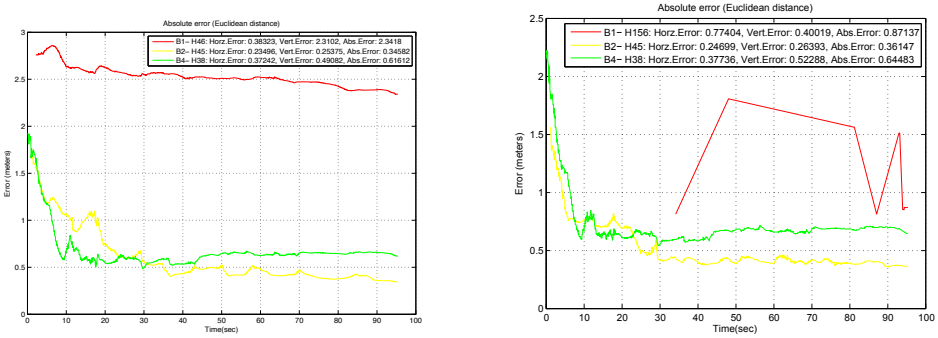


Figure 6.9: Evolution of the landmark 3D localization error with respect ground-truth for UAV1 in single configuration (left) and decentralized fusion (right).

6.5 Summary and conclusions

This chapter presented an approach for map fusion using centralized EKF-SLAM framework for each local SLAM problem in a multi-robot scenario. The multi-SLAM problem was solved with a decentralized approach but keeping the centralized state vector of robot position and map in order to allow to model the correlations between landmarks. Two methods were proposed for this decentralized multi-SLAM problem.

The first algorithm proposed uses the remote map information to prune local hypotheses of a landmark which do not intersect with the external probability distribution of the same landmark. This algorithm allows a faster convergence of the filter by pruning non overlapping hypotheses between different robot beliefs. This algorithm avoids the rumor propagation by not including the external map belief information in the local filter, instead this external map belief is just used to prune local hypotheses.

The second algorithm extends the first approach by including in the local landmark belief the intersecting hypotheses coming from the external robot but ponderating the local hypotheses over external ones to avoid inconsistencies in the EKF filter due to rumor propagation problem. then, this algorithm avoids rumor propagation by using a convex combination of maps similar to the approach followed in Covariance Intersection. In practice, the inclusion of external hypotheses in local filter acts similar to the re-sampling process of particles filters giving better results than first method since it allows to improve the coverage of

the most probable landmarks positions.

Experimental results showed that the improvement in RO-SLAM estimation is less significant than in simulation, because real range-only sensors introduce larger errors. The proposed approach always provided better results and significantly reduced in all cases the number of hypotheses per landmark, reducing the computational requirements.

Chapter 7

Active perception for 3D RO-SLAM

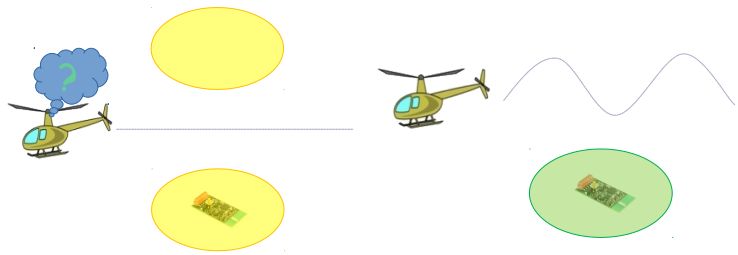
7.1 Introduction

Most of localization approaches do not take into account the possibility of controlling the robot to improve the perception, instead, the robot is just commanded with a predefined path. Active sensing strategies may lead to more efficient exploration and mapping approaches. The robot can adapt its trajectory, avoiding for instance non-observable motions or following those paths which are most informative. Thus, in the case of RO-SLAM, this means selecting the path which is likely to produce the highest reduction in the uncertainty on the nodes' positions.

7.2 Overview and related work

Active sensing approaches are especially well suited for ill-posed estimators such as RO-SLAM in the early steps of the estimation where perception significantly depends on the robot actions. In this case, there are many possible localization hypotheses that can only be discarded based on the robot trilateration at different positions. This thesis solves the problem by using a centralized Extended Kalman Filter presented in chapter 3 that stores all possible localization hypotheses and updates their estimation as soon the robot moves to a different position. This approach is undelayed, so the measurements can be integrated into the filter since the very first range data. However, the time required for the filter to converge to a single localization solution for each landmark will always depend on the robot trilateration no matter the approach. Actually there is a chance the estimation will never converge to the correct position if the robot trilaterates very poorly.

Figure 7.1 illustrates the benefit of considering active sensing strategies for RO-SLAM. Thus, the robot trajectory of Figure 7.1a results in two hypotheses with very similar uncertainty (bimodal distribution of the node position). On



(a) (b)

Figure 7.1: Two examples of range-only localization. The aerial robot receives range data from the beacon at different positions. Ellipses denote estimations over landmark position. (a) Example of vertical flip ambiguity with constant altitude trajectory. (b) Solved ambiguity when using a sinusoidal vertical trajectory.

the other hand, Figure 7.1b shows how adapting the robot trajectory benefits the localization of the node.

Active perception techniques requires a metric about the amount of uncertainty reduced when selecting a certain task or action. One of the most common metrics is the gain of information, for Bayesian approaches this gain might be computed as the (expected) variation on the entropy of the beliefs on landmarks' position. An example of this active sensing approach is used in [26], [91] or [12, 90] for exploration and SLAM. In [108], active sensing strategies are applied to the problem of tracking using only range measurements, where the target is represented by a single Gaussian. In [101] it is presented an active perception approach which computes the amount of uncertainty that would result after applying different available robot actions. This amount of uncertainty is measured by analyzing the eigen values of the new estimated covariance matrix at each the expected robot location after each candidate action.

In this chapter an active sensing approach is presented in order to maximize the gain of information while the aerial robot moves between waypoints. The method extends a previous work [75] with a full 3D RO-SLAM for aerial robots. The approach is based on the selection of robot actions that allows maximizing the gain of information while the robot is moving. In order to integrate the active perception with a local planner a loose coupling has been selected. The approach has been implemented in C++ under ROS and validated in simulations using a

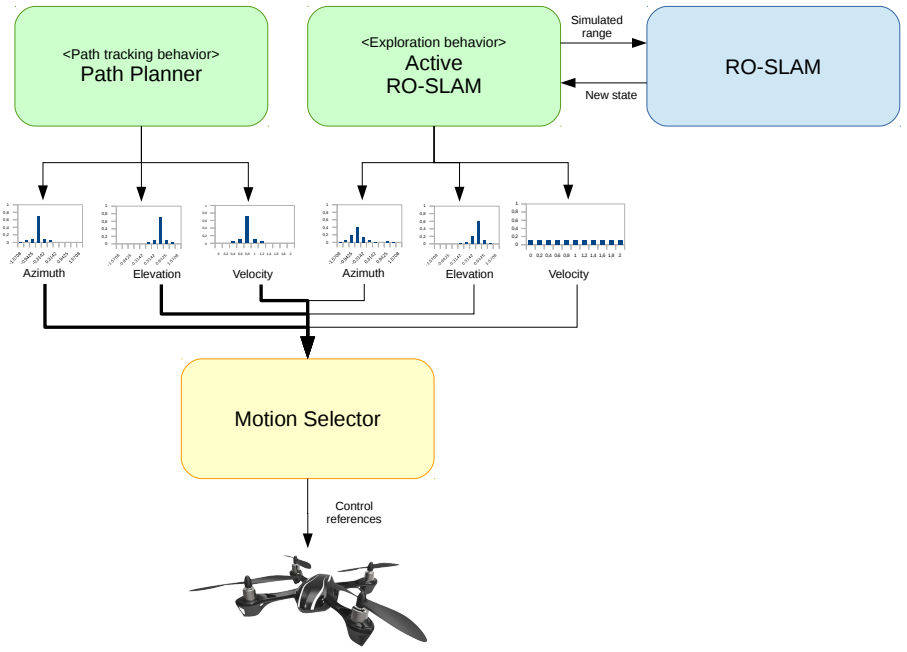


Figure 7.2: Combination of behaviors path tracking against exploration behavior using a loose coupling system. The actions of the path planner are weighted with a higher relevance (see thickness of the arrows) with respect active perception actions, making the system follow the initial planned path but with small variations introduced by the exploration behavior used to reduce RO-SLAM uncertainty.

3D environment simulator.

7.3 Active sensing for improved 3D RO-SLAM

The benefit of using an aerial robot to estimate the position of a set of landmarks is the possibility to control its motion in order to follow the most informative path. On the one hand, from the set of possible motions of the robot, it should take those that allows to estimate the position of the landmarks more accurately. On the other hand, the robot should try to avoid motions that decrease the observability of the landmark position.

When using active perception techniques for a particular aerial robot mission, a common approach is to use a combination of behaviors as shown in Figure 7.2. These systems are based on a combination of tasks, each of them with an associated cost function and a normalized behavior weight which allows the system to select the action which best fit the requirements of all behaviors according to their priority. In this chapter, two major tasks are considered: the most important task is the tracking of a path given by a local motion planner, the second task is the exploration of the environment in order to map the position of a set of radio beacons or landmarks. For the exploration of the environment, the system computes the gain of information when using a particular action. This gain is estimated using the expected variation of entropy of the RO-SLAM covariance matrix.

7.3.1 Entropy-based active sensing strategy

The key idea is to select the action that maximize the reduction in entropy in the centralized EKF covariance matrix. Thus, the robot should be actively moved to gain as much information as possible. In order to do this, it is needed to define a measurement of the information gain obtained when executing a certain exploration action.

A common metric about the information of a probability distribution is its associated entropy. The entropy H of a probability distribution $p(x)$ is defined as the expected value of the information $-\log[p(x)]$:

$$H(p(x)) = E_x[-\log p(x)] = -\int p(x)\log p(x)dx \tag{7.1}$$

With this entropy definition, the information gain is defined as the variation in the entropy of the distribution after carrying a certain action $\mathbf{u}_t$. After the execution of this action, the new distribution $p(\mathbf{x}_{t+\Delta t}|\mathbf{u}_t, \mathbf{z}_{t+\Delta t})$ is obtained from the future measurement $\mathbf{z}_{t+\Delta t}$ with an associated new entropy value denoted by $H(p(\mathbf{x}_{t+\Delta t}|\mathbf{z}_{t+\Delta t}, \mathbf{u}_t))$.

Then, as the only parameter which can be controlled is $\mathbf{u}_t$, the expected entropy should be computed for all potential measurements $\mathbf{z}_{t+\Delta t}$ obtained from this action. Therefore, the expected information gain associated to action $\mathbf{u}_t$ is defined as:

$$\Delta(\mathbf{u}_t) = H(p(\mathbf{x}_t)) - E_{\mathbf{z}_{t+\Delta t}}[H(p(\mathbf{x}_{t+\Delta t}|\mathbf{z}_{t+\Delta t}, \mathbf{u}_t))] \tag{7.2}$$

This metric can be used to chose the action that maximizes the value $\Delta(\mathbf{u}_t)$.

Entropy of a Gaussian Mixture

The entropy, as defined in equation (7.1), can be obtained analytically for certain distributions, including the Gaussian distribution. However, there is no analytical solution for the case of Gaussian Mixtures, defined by (3.1).

One option is to numerically integrate (7.1), for instance using Monte Carlo methods. However, this is computationally demanding, as a high number of samples may be required (the accuracy depends on the number of samples). The proposed approach uses upper bounds of the entropy as an approximation to the actual entropy value. Thus, instead of analyzing the expected variation using the analytical solution for a particular action, the expected variation of the entropy bound will be considered.

In [59], an analytical solution is derived to the Gaussian Mixtures entropy along to an upper and lower bound approximation. For active sensing approaches it is of particular interest the upper bound of the entropy, which might be computed for a Gaussian Mixture in a very cheap way as:

$$H(f(\mathbf{x})) \leq \sum_i \omega_i(-\log \omega_i + \frac{1}{2}\log((2\pi e)^N|\Sigma_i|)) \tag{7.3}$$

for $\mathbf{x}$ of dimension N. Where Σ_i is the covariance of each Gaussian i with weight ω_i.

Moreover, this bound is exact when only one hypothesis remains, or when the hypotheses are separated. Therefore, a possible strategy is to compare actions taking into account how they affect not the entropy itself, but the upper bound. While in theory a decreasing in the bound could not reflect on a decreasing of the actual entropy, in the experiment section it will be seen that the procedure is effective reducing the actual entropy of the distributions.

7.3.2 Active sensing architecture

The robot considered here is an aerial vehicle. The variables controlled by the local planner are the linear velocity v of the robot, the azimuth angle θ and the elevation angle ϕ. As previously introduced, the robot makes use of a loose coupling system able to merge two or more motion behaviors. Each motion behavior send the motion votes that satisfy its objectives to a centralized arbiter

Algorithm 6: Active perception algorithm

Data: $p(\mathbf{x}_t)$, Δt
Result: $(\Delta(\theta_i), \Delta(\phi_j))$
1: $\Theta = \{\theta_1, \cdots, \theta_i, \cdots\}$ A set of M orientations
2: $\Phi = \{\phi_1, \cdots, \phi_j, \cdots\}$ A set of N orientations
3: $H_t \leftarrow$entropy $(p(\mathbf{x}_t))$
4: **for all** $\theta_i \in \Theta$ **do**
5: **for all** $\phi_j \in \Phi$ **do**
6: $\mathbf{r}_{t+\Delta t} \leftarrow$ predict_robot $(\mathbf{r}_t, \theta_i, \phi_j, \Delta t)$
7: **for all** (μ_k, Σ_k) in $f(\mathbf{x}_t)$ **do**
8: $\mathbf{z}_{t+\Delta t} \leftarrow$ simulate_measurement$(\mathbf{r}_{t+\Delta t}, \mu_k, \Sigma_k)$
9: $p(\mathbf{x}_{t+\Delta t}|\mathbf{z}_{t+\Delta t}) \leftarrow$ update $(p(\mathbf{x}_t), \mathbf{z}_{t+\Delta t})$
10: $H_{i,j,k} \leftarrow$ entropy $(p(\mathbf{x}_{t+\Delta t}|\mathbf{z}_{t+\Delta t}))$
11: **end for**
12: $\Delta H \leftarrow H_t - \frac{\sum_k H_{i,j,k}}{K}$
13: $\Delta(\theta_i) \leftarrow \Delta(\theta_i) + \Delta H$
14: $\Delta(\phi_j) \leftarrow \Delta(\phi_j) + \Delta H$
15: **end for**
16: **end for**
17: Normalize $\Delta(\theta_i)$
18: Normalize $\Delta(\phi_j)$

(as in [87]) that merges them and take the action that better satisfy all the behaviors in the system. This combination associates a set of weights for all the potential contributions of the different behaviors.

Algorithm 6 shows the strategy to compute the votes associated to the active sensing behavior. Only the azimuth θ and the elevation ϕ angles will be considered, which are discretized into a set of M azimuth values $\{\theta_1, \cdots, \theta_L\}$ and N elevation values $\{\phi_1, \cdots, \phi_N\}$. For each potential angle (θ_i, ϕ_j) it is possible to predict the future position of the robot for a certain time horizon Δt. At that future position, the potential range measurements to the known nodes are considered. The basis of the algorithm is given by lines 8, 9 and 10. Within the for loop, each hypothesis within the Gaussian mixtures about the position of the known nodes is considered correct, and an artificial measurement $\mathbf{z}_{t+\Delta t}$ is simulated for that hypothesis at line 8. Then, the filter described in chapter 3 is applied by

the function `update` to estimate the future belief, and the upper bound of the entropy (7.3) is computed.

The final expected information gain is computed as the mean of these entropies. That is, taking the expectation with respect to all the potential measurements, which corresponds to the second term of the right hand side of (7.2).

Although not depicted in Algorithm 6, the final algorithm applies the same operation for all the currently known beacons that are within communication range. Therefore, the final vote $\Delta(\theta_i)$ and $\Delta(\phi_j)$ for a particular action is the sum of the variations of the entropy for each of these beacons.

The final votes for all elevation and azimuth angles are normalized. These votes are then combined with the votes indicated by other behaviors. Figure 7.3 shows an example of particular interest. It shows how the strategy not only can lead to reductions on the uncertainty, but also to avoid non-observable motions, like straight lines. In this example, it can be seen how there are two symmetric entropy variation maxima.

7.4 Results

The approach presented in this chapter is tested in simulation, where it can be assured that sensor data and robot actions can be under control. The whole active sensing architecture has been implemented in C++ using ROS. Actually, the same sensors used in datasets of appendix B are used, but with the difference that sensor data are artificially created based on the known position of the robot and the range sensors. On the other hand, UAV dynamics are emulated based on ROS-GAZEBO 3D models. Range-only sensor data are modeled taking into account outliers, Gaussian noise, bias, etc.

For this particular implementation the prediction interval Δt has been set to $2s$. Small values of Δt are discarded because landmark trilateration is almost not affected (for medium aerial robot velocities) when robot traverses short distances. On the other hand, much longer periods result in inaccuracies due to EKF linearizations. Thus, $\Delta t = 2s$ is a compromise between efficiency and accuracy.

The aerial robot was commanded to follow a given trajectory based on waypoints. As previously presented, the active approach will modify the desired elevation and azimuth angles in order to maximize the gain of information. The trajectory commanded to the UAV and the active actions resulted from the ap-

proach are presented in Figure 7.3. It can be seen how the active perception systems tends to move the UAV around the predefined path in order to gather more information of the range landmarks. From a theoretical point of view, the active perception system should force the robot to move to positions that increase the trilateration with respect the sensor landmarks.

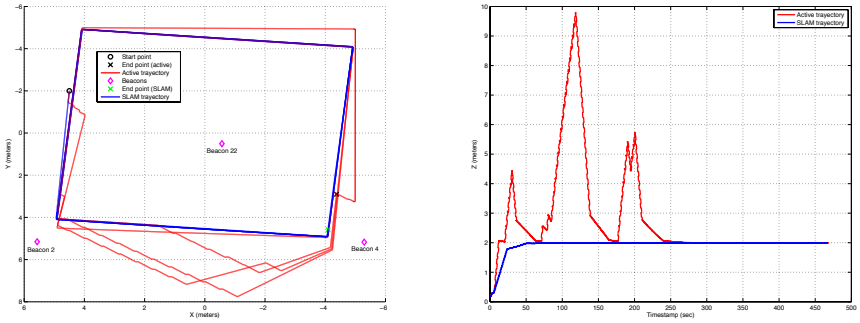


Figure 7.3: UAV trajectory with (red) and without (blue) active perception. (Left) XY trajectory. (Right) Z trajectory

The improved trajectory resulted in a better trilateration of the sensor nodes and, hence, better localization of the map. Figure 7.4 shows the estimated node position error with respect the ground-truth when active perception is considered and compared without it. It can be seen in the figure how the average error in the node position is clearly reduced, although the individual errors of some node position are larger without active perception. This effect is mainly produced by the approach itself that tries to improve the information gain globally, which might end with higher errors for individual nodes in favor of a better global map error.

The active perception approach has also impact in the convergence time needed by the SLAM approach to reach single hypothesis representation for each beacon in the map. This is consistent with the theory because a better trilateration helps to remove inconsistent hypotheses and, hence, it should provide faster convergence time. The evolution on the number of beacons hypotheses in the SLAM filter is shown in Figure 7.5 with and without active perception. It can be seen how the number of hypotheses converges to single faster with active

perception, the average convergence time without active perception is 40s (since the node is discovered) while with active strategies it is reduced to 26s.

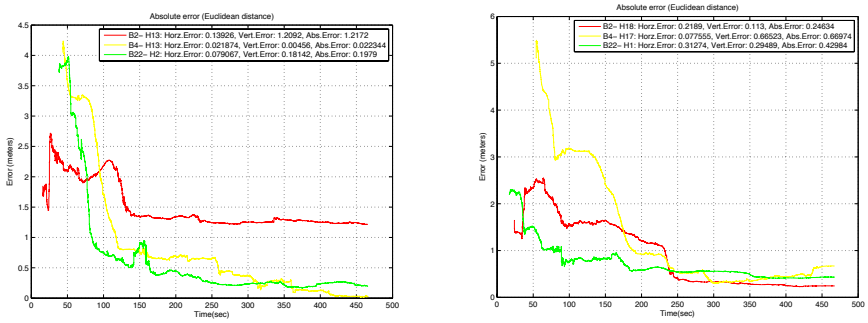


Figure 7.4: Evolution of the absolute localization error of every sensor node in the experiment. (Left) Estimation without active perception (Right) Estimation with active perception

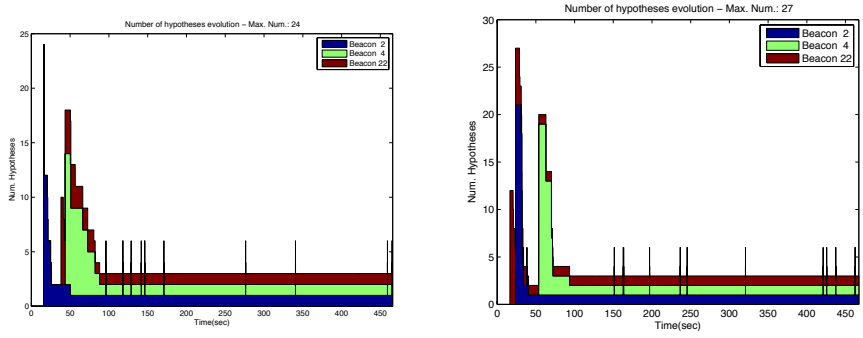


Figure 7.5: Evolution of the total number of beacon hypotheses with time. Results converge to 3 (one hypothesis per beacon). (Left) Results without active sensing. (Right) Results with active sensing)

7.5 Summary and conclusions

Most of localization approaches do not take into account the possibility of controlling the robot to improve the perception, instead, the robot is just commanded with a predefined path. This chapter presented an active perception approach based on a loose coupling system used to fuse different robot behaviors. In this case the system is based on path tracking and exploration behaviors.

The active perception technique proposed in this chapter is aimed to reduce the uncertainty of the map belief. The method is based on an action selection technique that uses the entropy of the SLAM belief to compute the estimated gain of information which turns out from each possible action. The set of actions considered in this approach are based on a fixed speed and a set of azimuth and elevation velocity angles between which the system must take a decision to reduce the map uncertainty.

In order to compute the gain of information, the system needs to compute the entropy of landmarks belief. However, this entropy is computationally inefficient for Gaussian Mixture distributions. For this reason, this chapter proposes to use a upper bound of the entropy which is sufficient to select the action with a higher gain of information.

The chapter ended showing simulation results of the method. These results proved how the system improves the convergence of hypotheses while at the same time allows to get a better accuracy in the mapping results.

Chapter 8

Conclusions and future work

8.1 Concusions

The application of simultaneous localization and mapping (SLAM) is a critical issue in aerial autonomous systems. The most common solution is to solve the localization problem with a GPS, inertial sensors and odometers. In the case of GPS, the signal is unavailable in indoors and sometimes in outdoor environments. The application of SLAM techniques with cameras and LIDAR sensors has become another common solution to this problem which might be applied even when the GPS signal is not available. However these sensors present several limitations when trading with applications where there is not a direct line of sight between the vehicle and the landmarks of the map. Some of these applications can be found for example in rescue scenarios, where there is no direct line of sight with some artificial of natural landmarks of the environment. Moreover, in these scenarios there are occlusions of static elements, such as walls, that can be used for SLAM by means of LIDAR sensors.

Range-only SLAM (RO-SLAM) is related with those algorithms based on the use of just range-only observations to map the unknown position of a set of range-only sensors (landmarks), while at the same time the robot gets localized with respect that map of sensors. This is a huge challenge since these sensors provides low-informative data, only composed by the distance between the robot and each landmark.

Different solutions are provided for 2D RO-SLAM in the literature. However, due to the rank-deficiency of the range-only observation model, the application of RO-SLAM to 3D scenarios becomes more challenging since the number of hidden variables increase with the dimensionality of the map. Thus, while in 2D RO-SLAM only the azimuth angle is the hidden variable, in 3D RO-SLAM a new hidden variable needs to be estimated increasing the computational requirements of the problem. This is different from bearing-only SLAM using monocular

cameras where the only hidden parameter is the distance between the sensor and landmarks when working in both 2D and 3D mapping problems (i.e. there is no increment of computational requirements in this case when increasing the dimensionality).

The first chapter of this thesis provided an extended introduction to Range-only SLAM problem and reviewed the state of the art of the problem. Later, some applications to range-only SLAM were proposed as the main framework of the development of the thesis. The main motivation found on this thesis is related with those applications which uses inter-landmark observations to improve the mapping estimation and those applications in which there is a known relative position constraint between landmarks, for example, when they are integrated in structural elements to be mapped and manipulated as is the case of the ARCAS project.

The first solution proposed to solve 3D RO-SLAM problem in chapter 2 was based on an Extended Kalman Filter with a delayed landmark state initialization which uses a non-parametric Bayesian filter to get an initial landmark estimation to be integrated in the EKF. This method was implemented in this chapter for 3D application as the baseline algorithm mostly employed in the literature to solve 2D RO-SLAM problem in order to compare the benefits and drawbacks of this method against undelayed multi-hypotheses approaches. The delayed initialization algorithm uses a particle filter for each new landmark. This algorithm models the initial landmark position belief using Monte Carlo sampling methods around the initial probability distribution of the range-only observation model. Once the landmark belief converges to a Gaussian distribution, this Gaussian distribution is appended to the global EKF state in which the position of the robot and other previously initialized landmarks are being estimated. The main drawback off this method is its bad scalability and the delayed integration of range-only measurements in the EKF state vector which does not allow to correct the position of the robot while no initialized landmarks are integrated in the state vector. On the other hand, experimental results showed that the convergence of this particles filters in 3D problems is very slow and sometimes inaccurate, leading the EKF estimation of the whole SLAM approach to divergences on the estimation.

In chapter 3 a new approach was presented using an EKF undelayed algorithm which uses Gaussian Mixture Models to model the initial landmarks distribution using multiple hypotheses. The solution implemented is based on the classic

EKF-SLAM, thus an initial estimation is given for the robot position when starting the algorithm. The main problem here is the initialization of landmarks in the state vector of the EKF. The initial non-Gaussian distribution of the landmarks position does not allow its direct integration in the EKF state. Two Gaussian mixture models are employed in this solution to estimate the azimuth and the elevation angles of the real position of the landmark. The main contribution of this solution was the use of a reduced spherical parametrization, which uses only $4 + N + M$ parameters for each landmark, where N is the number of samples employed to model the possible values of azimuth angle θ and M is the number of elevation angles ϕ sampled. The chapter also presented new observation models which are more scalable with the number of hypotheses and landmarks. The main improvement of these observation models consisted on a reduction on the number of equations required to update each azimuth and elevation hypothesis. The first method is based on common Federated Information Sharing approach but reducing the number of equations from $N \times M$ used in the literature to $N + M$ equations. The second used a single equation which does not require to split the variance of the range-only observations as is the case of Federated Information Sharing and also allowed the integration of hypotheses weights in the correction equations. Additionally, the chapter described how the second observation model can be applied to either, the reduced parameterization or one of the parameterizations used in the literature. To update the weights of hypotheses, two new efficient strategies were proposed in this chapter. These strategies are based on a conditional independence between azimuth and elevation angles allowing to reduce the number of weights to be computed and stored in memory from $N \times M$ to $N + M$. On the other hand, in order to reduce the computational resources employed, these parameters are reduced as new measurements are received and hence some of them can be discarded when their probability falls considerably. Experimental results showed how the method presents similar results than classical approaches but reducing the computational requirements and making the method more scalable with respect the number of landmarks and hypotheses.

Another contribution of Chapter 3 was the integration of inter-landmark observations in this multi-hypotheses frameworks in an efficient way. The second observation model proposed allowed to integrate this kind of observations with just one equation. Experimental results showed how the integration of these inter-landmark observations reduced the convergence time of hypotheses and presented

accurate results not only in the mapping of landmarks but in localization too. The use of a EKF-SLAM approach allowed to model the cross-correlations between landmarks specially in those applications in which inter-landmark observations are integrated, making the whole mapping results more accurate.

The results of chapter 3 have been published in two conference papers [34,35], in a workshop [42] and in a journal which is under review [40]. Additionally, the methods described in this chapter led to two full documented libraries, one for general purpose probabilistic filtering and the second for range-only SLAM. These libraries will be released as an executable to be used with ROS framework and will also be published in OpenSLAM.org web page. The drivers and data messages developed for the radio-based range-only sensors have been documented and published ROS community as a general ROS stack [33] which includes two ROS packages [32,44].

Next chapter was aimed to make the previous approach more robust against range-only sensor outliers and noisy measurements. Thus, in chapter 4, a pre-filtering of range-only observations was proposed to detect and avoid sensor outliers and to reduced the noise of measurements. The pre-filtering of noisy or wrong measurements was performed using a set of heuristics which are dependent on the application constraints and other general heuristics. After these heuristics, the algorithm applies a median filter to reduce the noise of range-only observations. The detection of outliers was based on movement of the estimated robot position and the difference between the previous and current range-only observations. The movement of the robot was compared with the range-only observations difference taking into account not only the mean value but the variance of this values using Bhattacharya distance. The chapter also proposed a model for static and dynamic range-only bias estimation which allows to make the global estimations of the map more accurate. Experimental results showed the improvement on the accuracy of the SLAM results.

Chapter 5, presented the results of the SLAM approach when range-only observations are fused with other sensors. As an application of the ARCAS project for the localization of structural elements, this chapter showed how to use RO-SLAM approach as a coarse estimation of the structural elements position and how to improve it with the observation of visual markers placed on top of range-only sensors. The chapter showed the equations required to correct landmarks estimation with the reduced parametrization using camera observations. Additionally, the chapter proposed two different methods to update the weights

of the landmark hypotheses when camera observations are integrated. The first method keeps the most likely hypothesis pruning the rest of hypotheses of the filter considering a low error in the camera observations. The second method proposed is more conservative and just updates the weights of each hypotheses using visual observations, making the method more robust against camera outliers. Experimental validation showed the results of fusing not only this camera observations but also the integration of the GPS and an altimeter for the improvement on the localization of aerial robots. With respect the two methods proposed for visual markers integration, experiments showed similar results for both methods, being the first method more efficient since it keeps just one hypothesis when the first visual marker is integrated.

The results of chapter 5 were published in an international conference paper [36]. This chapter is also comming from a stay in the Institut de Robòtica i Informàtica industrial (IRI) at the Technical University of Catalonia (UPC) for integration experiments of the visual marker detector developed at IRI for deformable markers used in ARCAS project with the algorithms developed in this thesis.

Cooperative multi-SLAM methods were described in chapter 6. It proposed an efficient and easy to implement algorithm which fuses two estimations of the same landmark but from different robots. The algorithm computes the intersection of both landmark beliefs and prunes those hypotheses from local robot belief (the map of one robot) that fall out of this intersection. With those intersected hypotheses, two different approaches are proposed for the same algorithm: the first approach just updates the weights of the local hypotheses which are inside the belief intersection by normalizing them, the second integrates in local landmark belief the external hypotheses of the same landmark from the other robot and later updates the weights of all new hypotheses, but weighting local ones to avoid the problem of rumor propagation. Experimental results demonstrated better convergence, however, the second approach is more robust due to its similarity with the resampling algorithm of particle filters. The accuracy is similar in both approaches, but the first approximation tends to be more efficient.

The contents of this chapter were published in a conference paper [43] and in a journal article [45].

Finally, in chapter 7, an active perception algorithm was proposed to improve the convergence of the filter and to avoid the common flip ambiguity of RO-SLAM multi-hypotheses techniques. This flip ambiguity increases in 3D scenarios where

the additional dimensionality requires a movement along the Z axis to avoid the additional flip ambiguity in this axis. The active perception technique computes the gain of information for each possible robot action by computing an upper bound of the Gaussian Mixtures entropy. The possible actions selected for aerial robots in this case are based on the linear speed of the robot and the azimuth and elevation orientation of the velocity vector. The active perception algorithm was used as a exploration behavior in the whole aerial robot system, fusing it with a trajectory tracking behavior. To fuse these behaviors the thesis proposed a loss decoupling approach which weights each behavior according to the required priority. The algorithm is validated with simulated and real experimentation showing a faster convergence when using the proposed active perception technique. The contents of this chapter were published in an international conference paper [39] and is being extended for the Journal on Intelligent and Robotics Systems.

8.2 Future developments

The work presented in this thesis can be enhanced at different levels, so the following future work is proposed:

- Despite this work has improved the efficiency and the convergence time with respect the state of the art, the accuracy of the method is similar to other approaches. In the future other Gaussian filters will be tested to improve the numerical stability of the filter, including an non-linear iterative optimization EKF correction.

- Despite different prediction models have been developed and tested in simulation, due to the lack of a good odometry model and observations for real experimentation, most of the results of this thesis were obtained with a simple Random Walk prediction stage of the Extended Kalman Filter (EKF). Future work will consider other odometry estimators using visual odometry or other optical flow sensors which are available in the market. This will improve the estimations of the EKF-SLAM algorithms proposed.

- The development of an information based filter has already been started and is under experimentation with some collaborations with the Institut de Robòtica i Informàtica industrial (IRI) at the Technical University of Catalonia (UPC). This information based filter will allow to improve the

active and cooperative perception techniques presented in this thesis by easing the computation of the entropy or by using Covariance Intersection techniques for map fusion.

- The Gaussian Mixture Correction method presented in this thesis might lead to bad linearization points due to the use of the initial Gaussian Mixtures expectation. Future developments will study different methods to guide the expectation value to the direction of movement of the vehicle. This idea is based on the assumption that new beacons are usually discovered while the robot moves, hence the azimuth and elevation angle hypotheses can be pruned to those angles which are in the opposite direction of the robot movement, making the initial expectation being oriented to the direction of movement.

- New techniques for active perception will also be studied based on The Rapidly-Exploring Random Tree (RRT) algorithm. The computation of the gain of information can also be used in RRT as a objective function in a local path planner which takes also into account the exploration behavior instead of using the Distributed Architecture for Mobile Navigation (DAMN) proposed in this thesis.

Finally, the libraries developed during this thesis are being licensed and documented in order to make them public and available for the research community under ROS framework and also for OpenSLAM community.

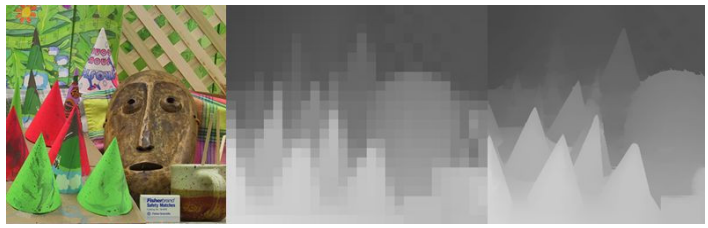
Appendix A

Introduction to probabilistic filtering

A.1 Introduction

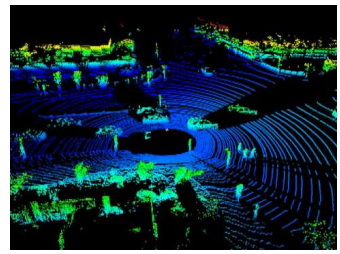
Probabilistic robotics has been a common approach used in robotics in the last decade. It pays tribute to the uncertainty in robot perception and motion instead of relaying on a single best approximation like other classical batch algorithms. This uncertainty, as stated in [5], comes from the incompleteness of robot perception and the inherently unpredictable environments which are present even in structured environments. As it's not possible to represent all variables of the environment, there are always hidden variables not taken into account in observation and motion models. As a result of these hidden variables, there isn't always a good match between models and real world. The basis of probabilistic robotics is the use of probabilistic theory to represent the uncertainty of the environment as well as the uncertainty related with perception and motion capabilities of robots. These probabilistic paradigms will allow to use maximum entropy principle to avoid the problem of incompleteness and converting it into uncertainty or estimations (i.e. a probability distributions), employing for that purpose a preliminary knowledge (motion and observation models).

Sensors are limited in what they can perceive. In that sense, incompleteness of robot sensors arises from several factors like sensor range or resolution. Figure A.1a shows a comparison of different resolutions incompleteness with a depth camera, while Figure A.1b shows how the depth camera has a limited depth perception (i.e. black background represents a lack of depth perception). In Figure A.1c is shown a different type of sensor incompleteness caused by partial occlusions (non direct Line Of Sight - NLOS) due to obstacles in front of the 3D LIDAR laser. Uncertainty is more related to calibration, noise or planning problems. In the same way, robot motion is limited by mechanical constraints



(a) Resolution incompleteness

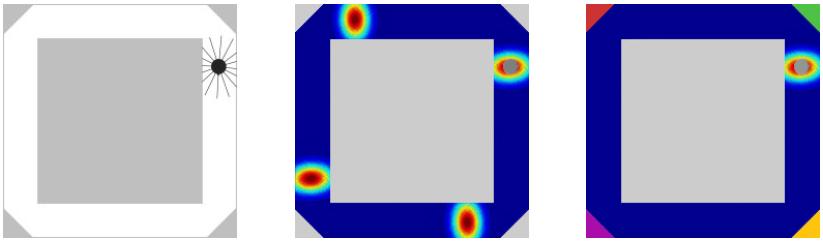
(b) Depth incompleteness



(c) NLOS incompleteness

Figure A.1: Sensor incompleteness

and noise and the main source of uncertainty comes from simplified models used for prior estimation of robot state. Figure A.2a shows an example of uncertainty where the robot can not disambiguate it's position by only employing a LIDAR sensor. The measures received in that well-structured environment are exactly the same in 4 positions of the map (that uncertainty is transformed from limitations - incompleteness - on what can be perceived by the sensor). The probabilistic theory gives some means to represent this uncertainty using different probabilistic distributions, this probability distribution represents what is called the belief of the robot. For example, Figure A.2b shows a multimodal gaussian distribution used to represent the uncertainty in robot position when only a LIDAR sensor is employed. As in other non probabilistic algorithms, it is possible to solve this ambiguous situations by merging the information of other sensors like in Figure A.2c, where the corners of the environment are painted with different colors and the robot uses a camera and some prior knowledge of the environment to disambiguate the ambiguous belief produced by LIDAR sensor.



(a) Map explored with LIDAR

(b) Robot belief

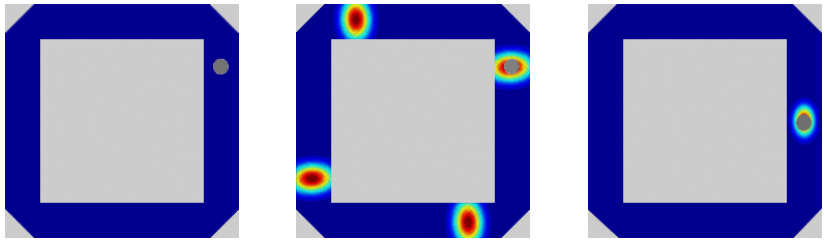
(c) Disambiguated belief

Figure A.2: Figure (a) shows an example of ambiguity generated by a structured environment built with a LIDAR sensor. In (b) a multimodal gaussian distribution is used to represent the belief of robot. In (c) shows a map with painted walls used to disambiguate the robot belief when a camera is used together with the LIDAR sensor.

Now, considering the case where the motion information is include into robot's belief as shown in Figure A.3. At first, in Figure A.3a, the robot is placed in an environment, but the robot doesn't know its initial position (in this example the map is known) so the probability distribution of robot belief is a uniform distribution (blue path). Then, the robot sense the environment with a LIDAR sensor (range-bearing measurements), as shown in Figure A.3b, so that the robot's belief is updated with 4 probable positions represented with a multimodal gaussian distribution. When robot moves down as depicted in Figure A.3c, the information about the action taken by the robot is included into robot's belief and, as the robot knows the map of the environment, it is able to disambiguate its position belief since only one of the 4 previous positions allows the robot to move in that way.

Stated probabilistically, the robot perception is a state estimation problem which can be solved with Bayesian filters. Bayesian filters attempts to update the robot belief employing the sensors and motion information. But, this update, can turn out into an information gain or into a loss of information depending on the dynamics of the robot's environment.

In contrast with classical solutions, probabilistic algorithms have weaker requirements on the accuracy of robotics sensors allowing to represent the degree of uncertainty of the robot by means of probabilistic distributions. Hence, proba-



(a) Uniform distribution (b) Posterior belief (c) Disambiguated belief

Figure A.3: Figure (a) shows the initial uniform distribution of a robot placed in the map. In (b), a LIDAR set of measurements are incorporated into robot's belief producing a mutimodal gaussian distribution. Finally, in (c), the robots disambiguate its position by incorporating motion information into robot's belief.

bilistic algorithms tend to be a more robust solution and scalable for real-world environments. However, the solutions proposed in this paradigm are generally more complex computationally since they not only offer a single solution but a complete probabilistic distribution. Indeed, in [22], the probabilistic paradigm has been demonstrated to be a NP-hard problem. The reason why researchers on this area are focused to provide solutions as efficient as possible specially when dealing with continuous state spaces, as is the case of this dissertation, where traditional solutions tend to be more efficient.

A.2 General concepts of probability theory

As stated above, in probabilistic robotics, the perception of the environment is represented by space states. Furthermore, all elements involved in robot's world like sensor measurements, actions, and the state of the robot are modeled as random variables. The value of random variables is enclosed in a specific domain which depends on the element of the robot's world they represent. These values are governed by probabilistic rules which must be inferred from other random variables and the observed information.

Let X denote a random variable which defines the domain of possible values x that this random variable can take. When the element represented depends on more than one value, the vector of these random variables is called a multivariate. To represent the probability of a single or multivariate random variable X to take

a value x, this document will use the notation $p(X = x)$, which will usually be abbreviated as $p(x)$.

On the other hand, another classification of random variables can be used depending on the variable domain. In that sense, the following types of random variables can be differentiated:

- **Continuous random variables:** These variables takes a continuous space domain and their probability distributions are represented by continuous probability density functions (PDFs). These PDFs must integrate to 1 when considering the entire domain:

$$\int_X p(x)dx = 1 \tag{A.1}$$

  In this thesis, the most common density function, is the normal or Gaussian distribution with mean μ and variance σ^2, abbreviated as $\mathcal{N}(x; \mu, \sigma^2)$ and which PDF is defined for a single random variable as:

$$p(X = x) = p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{A.2}$$

  For multivariate random variables, the normal distribution has the following PDF:

$$p(X = x) = p(x) = \frac{1}{\sqrt{2\pi\,|\Sigma|}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)} \tag{A.3}$$

- **Discrete random variables:** These variables takes a discrete space domain and their probabilistic distributions are represented by a discrete sum of the possible values of the random variable domain. Each value is bounded up to 1 and the probability distribution must sum up to 1:

$$\sum_X p(x) = 1 \tag{A.4}$$

  An example of discrete multivariate random variable might be the position of a robot in a 2D sampled space (grid) like in grid-based maps.

Another common term in probability theory and which is used in this dissertation is the joint probability. The joint probability, is the probability of a pair or more random variables to take a value in the domain of each random variable and is represented as follows:

$$p(X = x \wedge Y = y) = p(x \wedge y) = p(x, y) \tag{A.5}$$

Another important concept is known as the absolute independence, which refers to a set of variables which are completely independent between each other, and hence the joint probability can be divided in the product of independent probabilities as shown in the following equation:

$$p(x, y) = p(x)p(y) \tag{A.6}$$

A.2.1 Conditional probability

In robotics, sometimes it is needed to express that a variable carries information about other random variables. For example, the measurements of a LIDAR sensor carry information about robot's position. In those cases, this variables are told to be conditioned. An example of conditioning is the position of the robot, which is conditioned on the measures of a LIDAR sensor since, depending on the position of the robot, the LIDAR sensor will return a set of range-bearing measurements or others. The probability of a random variable X to take a value x, conditioned to a second random variable Y which value is y is denoted as follows:

$$p(x|y) = p(X = x|Y = y) \tag{A.7}$$

When $p(y) > 0$ the conditional probability is calculated as:

$$p(x|y) = \frac{p(x, y)}{p(y)} \tag{A.8}$$

Otherwise, for $p(y) = 0$, it is considered that $p(x|y)$ is undefined. On the other hand, if the random variables are independent, then the conditional probability $p(x|y)$ has the same value as if Y takes any value, i.e. $p(x|y) = p(x)$. This rule is derived from the property of absolute independence presented above (A.6):

$$p(x|y) = \frac{p(x, y)}{p(y)} = \frac{p(x)\cancel{p(y)}}{\cancel{p(y)}} = p(x) \tag{A.9}$$

Once some axioms of probabilistic theory have been defined, the following rules (the ones most used in probabilistic robotics) can be derived from them:

1. **Theorem of total probability:** This property comes from the axioms of probability and the rule (A.9), and is described by the following rules for discrete and continuous random variables respectively:

$$p(x) = \sum_{Y} p(x|y)p(y) \tag{A.10}$$

$$p(x) = \int_{Y} p(x|y)p(y)dy \tag{A.11}$$

   The product $p(x|y)p(y)$ in (A.10) and (A.11) is defined as 0 if either $p(x|y)$ or $p(y)$ are 0.

2. **Chain rule:** This property is derived from the basic conditional probability rule and allows to calculate the joint distribution of a set of random variables using only conditional probabilities. To explain this rule, consider the joint probability of a set of random variables $X_1...X_n$. The chain rule allow to calculate this probability as:

$$p(x_1, ..., x_n) = p(x_n|x_{n-1}, ..., x_1)p(x_{n-1}, ..., x_1)$$

   And, by repeating this rule recursively, the rule can be written as:

$$p\left(\cap_{k=1}^{n} x_k\right) = \prod_{k=1}^{n} p\left(x_k| \cap_{j=1}^{k-1} x_j\right) \tag{A.12}$$

3. **Bayes rule:** This is the most important rule in probabilistic robotics and in probabilistic inference in general, as it provides a rule to calculate $p(x|y)$ from its "inverse" conditional probability $p(y|x)$. As in (A.7), this rule requires $p(y) > 0$. The rule for discrete and continuous random variables is as follows:

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)} = \frac{p(y|x)p(x)}{\sum_{X'} p(y|x')p(x')} \tag{A.13}$$

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)} = \frac{p(y|x)p(x)}{\int_{X'} p(y|x')p(x')dx} \tag{A.14}$$

In equation (A.13) and (A.14), as $p(y)^{-1}$ not depends on value x, this rule is often written in normalized form:

$$p(x|y) = \eta p(y|x)p(x) \tag{A.15}$$

In this notation, η refers to a normalization factor which avoids the calculation of $p(y)$ and implies that the result of equation (A.15) should be normalized to 1.

It is possible to condition the Bayes rules to more than one random variable, thus, for example, for two conditional random variables Y and Z the Bayes rule would be expressed as:

$$p(x|y,z) = \frac{p(y|x,z)p(x|z)}{p(y|z)} = \frac{p(z|x,y)p(x|y)}{p(z|y)} \tag{A.16}$$

4. **Conditional independence:** This rule extends the conditional independence for joint probabilities conditioned to a set of random variables and its application is very similar to (A.6):

$$p(x,y|z) \underbrace{=}_{Eq.A.7} \frac{p(x,y,z)}{p(z)} \underbrace{=}_{Eq.A.12} \frac{\cancel{p(x|y,z)}\overset{p(x|z)}{\overbrace{\hphantom{p(x|y,z)}}}p(y,z)}{p(z)} \underbrace{=}_{Eq.A.12} p(x|z)p(y|z) \tag{A.17}$$

Despite the rules for absolute and conditional independence are similar, conditional independence does not imply absolute independence (A.6) and vice versa. That means that two variables can be jointly conditionally dependent to a variable Z and at the same time this two variables can be be independent between each other (A.6). However, in some cases, both kind of independence might meet.

A.2.2 Expectation, variance and entropy

The algorithms used in probabilistic robotics require to compute a set of statistics from probability distributions. The most important statistics in probabilistic robotics are the expectation, covariance and entropy. This statistics are described in the following subsections.

- **Expectation:** The expectation is the expected value of a random variable if the process is repeated infinitely and is calculated as the weighted mean value of all possible values of the random distribution. This weighted mean value is calculated for the case of discrete and continuous random variables as follows:

$$E[X] = \sum_X xp(x) \tag{A.18}$$

$$E[X] = \int_X xp(x)dx \tag{A.19}$$

  An important property of the expectation is its linearity with respect the random variables. Suppose a and $b \in \mathcal{R}$, then:

$$E[aX + b] = aE[X] + b \tag{A.20}$$

- **Variance and covariance:** The variance σ^2 is a single value which measures the squared expected deviation σ of a single random variable from the mean value obtained with the expectation statistic, whilst the covariance $\mathbf{\Sigma}$ is matrix used for multivariate probabilities and calculates not only the variance of each individual random variable but also the correlation between each pair of variables. The following matrix represents the covariance matrix of two random variables, X and Y.

$$\mathbf{\Sigma}(X,Y) = E[(X - E[X])(Y - E[Y])] \tag{A.21}$$

$$\Sigma(X,Y) = \begin{bmatrix} \sigma^2(X) & cov(X,Y) \\ cov(Y,X) & \sigma^2(Y) \end{bmatrix}$$

where the cross-covariances are computed as $cov(X,Y) = E[(X - E[X])(Y - E[Y])^T]$ and $cov(Y,X) = E[(Y - E[Y])(X - E[X])^T]$.

The correlation between two variables $corr(x,y) = \frac{cov(X,Y)}{\sigma(X)\sigma(Y)}$, indicates how a random variable is affected by a change in the other. Thus, if the correlation between two variables is positive, then both variables change in the same way, but if the correlation is negative, an increase in a random variable will suppose a decrease on the other. When two random variables are independent, then they are uncorrelated, which means that their cross-covariance value is zero. With this properties, one may notice that the variance is a special case of covariance for a single random variable.

The intrinsic properties of a covariance matrix are:

1. **Bilinear:** for constants a and b and random variables X, Y, Z, $\sigma(ax+by,z) = a\sigma(x,z) + b\sigma(y,z)$
2. **Symmetric:** $\sigma(x,y) = \sigma(y,x)$
3. **Positive semi-definite matrix:** $\sigma^2(x) = \sigma(x,x) \geq 0$ for all random variables X, and $\sigma(x,x) = 0$ implies that X is a constant random variable.

When a vector of random variables (or multivariate) $\boldsymbol{x}$ is transformed by a linear transformation A, the covariance matrix is then transformed according to the following equation due to its derivation of the expectation statistic which is linear too.

$$\boldsymbol{\Sigma}(A\boldsymbol{x}) = A\boldsymbol{\Sigma}(\boldsymbol{x})A^T \tag{A.22}$$

- **Entropy:** The entropy H originates in information theory, and is a measure of unpredictability or information content:

$$H(X) = E[-log_2 P(X)] \tag{A.23}$$

This expectation resolves for discrete and continuous random variables as follows:

$$H(X) = -\sum_X p(x) log_2 p(x) \tag{A.24}$$

$$H(X) = -\int_X p(x) log_2 p(x) dx \tag{A.25}$$

Then, the entropy suppose a good statistic to estimate the gain of information when a robot takes a specific action. Higher values of entropy indicates higher uncertainty, so decision making algorithms use this statistic to look for those actions that make the new entropy lower than the actual one, i.e. a gain of information.

A.3 Bayesian filters

Before describing the general Bayes filter algorithm it is necessary to introduce some basic concepts related with this general algorithm. Then the general Bayes filter algorithm is introduced to further explain Gaussian filters and Non-parametric filters as an implementation of Bayesian filters.

A.3.1 Concepts

One of the most important concepts related with Bayes filters is the concept of state. The state represents the actual characteristics of the environment including the robot characteristic like the position of the robot, the position of people around the robot, the weather or anything which can affect the objective tracked by the robot. States can be composed by dynamic or static elements. Dynamic elements are those which change over time (e.g. people, other cooperative or non-cooperative robots, etc), while static elements of the state are those which does not change but are relevant to accomplish the robot mission (e.g. static radio emitters, walls, etc). Those objects which are distinct, stationary features of the environment and hence can be recognized reliably are called landmarks, a typical example of a landmark is the visual one shown in Figure A.4 for an Augmented Reality (AR) application.

Following the Markov assumption, a complete state (the concept of complete state comes from process known as Markov chains) $\boldsymbol{x}^t$ is a state which contains the necessary information about past events and states in order to predict future

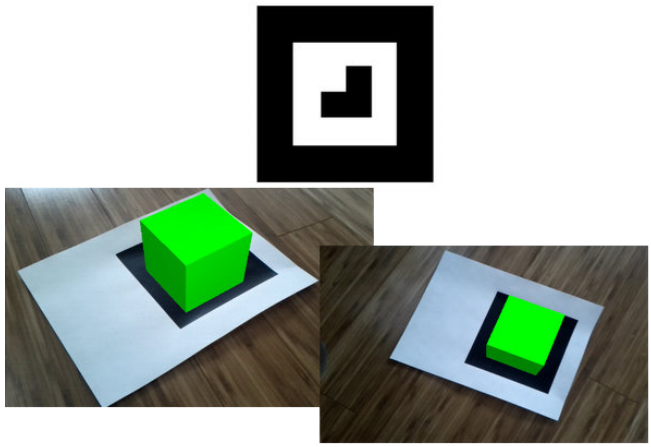


Figure A.4: Example of visual landmark.

states stochastically. In that way, a complete state $\boldsymbol{x}^t$ will be independent from past events like measurements $\boldsymbol{z}^{0:t-1}$, actions $\boldsymbol{u}^{0:t-1}$ and even, previous states $\boldsymbol{x}^{0:t-1}$.

Measurement vectors $\boldsymbol{z}$ carry information relative to perceived data from the environment from any sensor on-board robot or not. An example of measurement vector might be a set of n distance measurements $\boldsymbol{z}^t = [r_1, r_2, \ldots, r_n]^T$, which is indeed the most common vector state used in this dissertation. On the other hand, control vectors or motion vectors $\boldsymbol{u}$ carry information about changes produced in the state (people moving around, the motion of the robot, the motion of dynamic landmarks, etc). An example of motion vector might be the data provided by odometers in a ground mobile robot $\boldsymbol{u}^t = [v, w]^T$, where v is the linear velocity of the robot and w is the angular velocity of the robot.

Finally, one of the most important concepts in probabilistic robotics is the concept of belief. The belief of a robot is described as the internal knowledge of the current state of the robot, i.e. the belief represents the state of the robot with a probability distribution which is conditioned not only on what robot perceives but also in what robot do. As in [92], here $bel(\boldsymbol{x}^t)$ is used to represent the belief over the current state, which is an abbreviation of:

$$bel(\boldsymbol{x}^t) = p(\boldsymbol{x}^t | \boldsymbol{z}^{1:t}, \boldsymbol{u}^{1:t}) \tag{A.26}$$

A.3.2 State estimation

In probabilistic robotics, Bayes rule is the basic rule used in Bayesian filters and in probabilistic inference in general. Taking into account the Bayes rule (A.13), X might be a random variable representing a quantity to be estimated and Y the data used to infer X. In that rule $P(X)$ is known as the prior distribution, which summarizes the current knowledge of X prior incorporating the new data Y. The probability $p(x|y)$ is called the posterior probability distribution over X. The Bayes rule allows to infer the posterior probability through the probability $p(y|x)$ which is often called the generative model. As explained before, the probability $p(y)$ is not calculated, instead equation (A.15) is used.

When updating the state $\boldsymbol{x}^{0:t-1}$ to $\boldsymbol{x}^t$, generative laws of probabilistic robotics are used to calculate the posterior $p(\boldsymbol{x}^t|\boldsymbol{x}^{0:t}, \boldsymbol{z}^{1:t}, \boldsymbol{u}^{1:t})$. But, as stated above, assuming the Markov assumption, the posterior is independent from past states, measurements and actions, since $\boldsymbol{x}^t$ is complete and hence the posterior can be expressed as:

$$bel(\boldsymbol{x}^t) = p(\boldsymbol{x}^t|\boldsymbol{z}^t, \boldsymbol{u}^t) \tag{A.27}$$

To calculate this posterior, it might be useful to calculated a prior probability before incorporating measurement $\boldsymbol{z}^t$, thus a probability often referred as prediction is calculated as:

$$\bar{bel}(\boldsymbol{x}^t) = p(\boldsymbol{x}^t|\boldsymbol{u}^t) \tag{A.28}$$

The process of calculating the posterior belief from this prior belief is called correction or measurement update. Then, a generic Bayesian filter algorithm for continuous state is detailed in Algorithm 7.

Algorithm 7: Generic Bayesian filter algorithm

Input: $bel(\boldsymbol{x}^{t-1})$, $\boldsymbol{u}^t$ and $\boldsymbol{z}^t$
Output: $bel(\boldsymbol{x}^t)$

for all $\boldsymbol{x}^t$ **do**
  $\bar{bel}(\boldsymbol{x}^t) = \int p(\boldsymbol{x}^t|\boldsymbol{x}^{t-1}, \boldsymbol{u}^t) bel(\boldsymbol{x}^{t-1})$;
  $bel(\boldsymbol{x}^t) = p(\boldsymbol{z}^t|\boldsymbol{x}^t)\bar{bel}(\boldsymbol{x}^t)$;

The algorithm shows how the prediction stage is calculated using the theorem of total probability (A.11) by calculating the integral over the previous state of posterior (A.28) and the previous belief $bel(\boldsymbol{x}^{t-1})$. The prediction (A.28) depends on the motion model of the control values contained in the motion vector. Then, the predicted belief is used to integrate the measurement $\boldsymbol{z}^t$ and get the posterior belief $bel(\boldsymbol{x}^t)$. To calculate the probability $bel(\boldsymbol{x}^t) = p(\boldsymbol{x}^t|\boldsymbol{z}^t)$, the algorithm uses again the normalized Bayes rule (A.15) where $p(\boldsymbol{x}^t) = \bar{bel}(\boldsymbol{x}^t)$ as it is proved in [92]. The probability $p(\boldsymbol{z}^t|\boldsymbol{x}^t)$ is called the measurement probability and , as can be seen, depends on the actual state $\boldsymbol{x}^t$ which is a common characteristic of hidden Markov models (HMM) or dynamic Bayes networks (DBM). This probability is calculated from the measurement model of the sensor employed for each kind of measurement.

Another important aspect of this algorithm is that, as one may notice, is a recursive algorithm since it depends on the belief at previous state. This recursion implies that the initial belief $bel(\boldsymbol{x}_0)$ must be given. The initial value of the belief is one of the major challenges in RO-SLAM and hence, it is one of the mayor objectives of this thesis. The following algorithms will explain different general strategies to give the initial belief and the means to obtain the posterior probability according to the Bayes filter Algorithm 7.

A.3.3 Gaussian filters

Gaussian filters are the earliest and most common implementation of the Bayesian filter algorithm for continuous spaces. They are based on the representation of the belief by multivariate normal distributions (A.3). Then, Gaussian filters represent the belief with the common Gaussian parameters, i.e. the belief is represented by two parameters, the mean vector $\boldsymbol{\mu}$ and the covariance matrix $\boldsymbol{\Sigma}$. The mean $\boldsymbol{\mu}$ of these Gaussian filters is directly associated to the state vector $\boldsymbol{x}$, while the covariance matrix $\boldsymbol{\Sigma}$ is a matrix with the quadratic dimensionality of the state vector $\boldsymbol{x}$, and formed as described in section A.2.2 from the mean vector. The covariance $\boldsymbol{\Sigma}$ represents the degree of uncertainty of the belief for each parameter and has the advantage to keep the correlation between the different parameters of the state vector. This representation is known as the moments parametrization [1], but exists an alternative representation of this filters known

[1]The mean and covariance of a Gaussian distribution are the first and second moments of a probability distribution. Higher order moments are zero for Gaussian distributions.

as the canonical parametrization which are used for information filters. The transformation between canonical and moments parametrization is bijective.

However, Gaussian Filter requires to initiate the parameters of the filter (μ and Σ) with an initial estimation and assumes that the posterior of the belief can be represented with a unimodal Gaussian distribution, but this assumption doesn't fit with many localization applications where the robot has to localize itself from the scratch in an unknown environment (e.g. global and kidnapped robot problems).

In the following sections, the Kalman Filter (KF) and the Extended Kalman Filter (EKF) will be introduced as they are one of the most common and efficient implementations of Gaussian filters and Bayesian filters in general. Other implementations of Gaussian filters, not used in this thesis, are mentioned without giving many details.

The Kalman Filter - KF

The Kalman filter implements a Gaussian filter for continuous state spaces. This implementation is used as a filter or prediction employing linear Gaussian generative models. Hence, the generative models must be linear systems with added Gaussian noise.

For the motion model, this property is expressed by the following linear expression:

$$\boldsymbol{x}^t = \boldsymbol{A}^t \boldsymbol{x}^{t-1} + \boldsymbol{B}^t \boldsymbol{u}^t + \boldsymbol{\epsilon}^t \tag{A.29}$$

Which can be used to calculate $p(\boldsymbol{x}^t|\boldsymbol{u}^t, \boldsymbol{x}^{t-1})$ embedding (A.29) in (A.3):

$$p(\boldsymbol{x}^t|\boldsymbol{u}^t, \boldsymbol{x}^{t-1}) = \frac{1}{\sqrt{|2\pi \boldsymbol{R}^t|}} \mathrm{e}^{-\frac{1}{2}(\boldsymbol{x}^t - \boldsymbol{A}^t \boldsymbol{x}^{t-1} - \boldsymbol{B}^t \boldsymbol{u}^t)^T (\boldsymbol{R}^t)^{-1} (\boldsymbol{x}^t - \boldsymbol{A}^t \boldsymbol{x}^{t-1} - \boldsymbol{B}^t \boldsymbol{u}^t)} \tag{A.30}$$

Here, $\boldsymbol{A}^t$ and $\boldsymbol{B}^t$ are the matrices which characterize the motion model with dimensions nxn and nxm respectively, being n the dimensionality of the state vector $\boldsymbol{x}^t$ and m the dimensionality of the motion vector $\boldsymbol{u}^t$. This representation assumes that the motion model is a linear dynamic system with an added Gaussian noise $\boldsymbol{\epsilon}^t$ characterized by zero mean vector of the same dimensionality of $\boldsymbol{x}^t$ and covariance represented by the matrix $\boldsymbol{R}^t$.

For the observation model, the linearity is expressed by the following expression:

$$\boldsymbol{z}^t = \boldsymbol{C}^t \boldsymbol{x}^t + \delta^t \tag{A.31}$$

And hence, the measurement probability $p(\boldsymbol{z}^t|\boldsymbol{x}^t)$ becomes:

$$p(\boldsymbol{z}^t|\boldsymbol{x}^t) = \frac{1}{\sqrt{\left|2\pi \boldsymbol{Q}^t\right|}} \mathrm{e}^{-\frac{1}{2}(\boldsymbol{z}^t - \boldsymbol{C}^t\boldsymbol{x}^t)^T (\boldsymbol{Q}^t)^{-1} (\boldsymbol{z}^t - \boldsymbol{C}^t\boldsymbol{x}^t)} \tag{A.32}$$

Where the matrix $\boldsymbol{C}^t$ represents a linear observation model with an added Gaussian noise $\boldsymbol{\delta}^t$, parametrized with a zero mean vector of the same dimensionality of $\boldsymbol{z}^t$ and covariance $\boldsymbol{Q}^t$.

As described before, Gaussian filters require a initial belief and, in the case of Kalman filters, that means to give the initial belief as a vector with the initial expectation of the belief and a covariance matrix which represents the uncertainty of the initial expectation. Special care must be taken when initializing this kind of filters since, a huge initial covariance (i.e. huge degree of initial uncertainty), might lead the filter to diverge and make the covariance matrix to become inconsistent making the filter fail in real implementations.

A general algorithm for the Kalman filter is shown in Algorithm 8 as a pseudo-code algorithm. In that algorithm, given the current mean vector $\boldsymbol{\mu}^{t-1}$ and covariance $\boldsymbol{\Sigma}^{t-1}$, lines 1 and 2 represent the prediction stage of the algorithm, using the motion vector $\boldsymbol{u}^t$. The parameters of the transition Gaussian distribution $\bar{bel}(\boldsymbol{x}^t)$ are $\bar{\boldsymbol{\mu}}$ and $\bar{\boldsymbol{\Sigma}}$. These lines implements the equation (A.30).

Algorithm 8: Kalman filter algorithm

Input: $\boldsymbol{\mu}^{t-1}$, $\boldsymbol{\Sigma}^{t-1}$, $\boldsymbol{u}^t$ and $\boldsymbol{z}^t$
Output: $\boldsymbol{\mu}^t$ and $\boldsymbol{\Sigma}^t$

/*Prediction stage*/
1 $\bar{\boldsymbol{\mu}}^t = \boldsymbol{A}^t \boldsymbol{\mu}^{t-1} + \boldsymbol{B}^t \boldsymbol{u}^t$;
2 $\bar{\boldsymbol{\Sigma}}^t = \boldsymbol{A}^t \boldsymbol{\Sigma}^{t-1} (\boldsymbol{A}^t)^T + \boldsymbol{R}^t$;

/*Correction stage*/
3 $\boldsymbol{S} = \boldsymbol{C}^t \bar{\boldsymbol{\Sigma}}^t (\boldsymbol{C}^t)^T + \boldsymbol{Q}^t$;
4 $\boldsymbol{K}^t = \bar{\boldsymbol{\Sigma}}^t (\boldsymbol{C}^t)^T \boldsymbol{S}^{-1}$;
5 $\boldsymbol{\mu}^t = \bar{\boldsymbol{\mu}}^t + \boldsymbol{K}^t (\boldsymbol{z}^t - \boldsymbol{C}^t \bar{\boldsymbol{\mu}}^t)$;
6 $\boldsymbol{\Sigma}^t = (\boldsymbol{I} - \boldsymbol{K}^t \boldsymbol{C}^t) \bar{\boldsymbol{\Sigma}}^t$;

Then, the posterior $bel(\boldsymbol{x}^t)$ is calculated in lines 3 through 6 in the correction stage of the algorithm, incorporating the measurement vector $\boldsymbol{z}^t$. The new parameters of the posterior Gaussian distribution $bel(\boldsymbol{x}^t) = p(\boldsymbol{x}^t|\boldsymbol{u}^t, \boldsymbol{z}^t)$ are $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$. The implementation of Equation (A.32) uses a matrix $\boldsymbol{K}^t$, called Kalman Gain, and specifies the degree in which the new measurement vector is incorporated in the new state estimation. The mean vector is updated using the Kalman gain multiplied by the difference between the received measurement and the expected one (this difference is called the *innovation*).

The computational complexity of this algorithm depends on the number of parameters of the state vector and the number of parameters in the measurement vector. The solution is quite efficient and, actually, there are different new versions of the same algorithm which reduce the complexity of the algorithm for certain sparse updates (Sparse Information Filter).

The Extended Kalman Filter - EKF

The extended Kalman filter is an adaptation of the previous Kalman filter, specially designed for non-linear systems which are the most common in real applications. Here, the motion and/or the observation model are supposed to be non-linear and hence linearisation techniques are employed to approximate the belief into a Gaussian distribution. The difference between KF and EKF is that, while in KF the Gaussian distribution was exact, in EKF the Gaussian distribution is approximated. Now, the new generative models are:

$$\boldsymbol{x}^t = g(\boldsymbol{u}^t, \boldsymbol{x}^{t-1}) + \boldsymbol{\epsilon}^t \tag{A.33}$$

$$\boldsymbol{z}^t = h(\boldsymbol{x}^t) + \boldsymbol{\delta}^t \tag{A.34}$$

The matrices $\boldsymbol{A}^t$ and $\boldsymbol{B}^t$ have been replaced by the non-linear function $g(\boldsymbol{u}^t, \boldsymbol{x}^{t-1})$, and $h(\boldsymbol{x}^t)$ replaces matrix $\boldsymbol{C}^t$. The linearisation of $g(\boldsymbol{u}^t, \boldsymbol{x}^{t-1})$ and $h(\boldsymbol{x}^t)$ is often done with a method called (first order) Taylor expansion, which consist to calculate the Jacobian of $g(\boldsymbol{u}^t, \boldsymbol{x}^{t-1})$ at value $\boldsymbol{x}^{t-1} = \boldsymbol{\mu}^{t-1}$ and $h(\boldsymbol{x}^t)$ at value $\boldsymbol{x}^t = \bar{\boldsymbol{\mu}}^t$ to approximate both functions:

$$\boldsymbol{G}^t = g'(\boldsymbol{u}^t, \boldsymbol{\mu}^{t-1}) = \frac{\partial g(\boldsymbol{u}^t, \boldsymbol{\mu}^{t-1})}{\partial \boldsymbol{x}^t} \tag{A.35}$$

$$\boldsymbol{H}^t = h'(\bar{\boldsymbol{\mu}}^t) = \frac{\partial h(\bar{\boldsymbol{\mu}}^t)}{\partial \boldsymbol{x}^t} \tag{A.36}$$

Then, the approximation of function g and h at point $\boldsymbol{\mu}^t$ and $\bar{\boldsymbol{\mu}}^t$ respectively is:

$$g(\boldsymbol{u}^t, \boldsymbol{x}^{t-1}) \approx g(\boldsymbol{u}^t, \boldsymbol{x}^{t-1}) + \boldsymbol{G}^t(\boldsymbol{x}^{t-1} - \boldsymbol{\mu}^{t-1}) \tag{A.37}$$

$$h(\boldsymbol{x}^t) \approx h(\boldsymbol{x}^t) + \boldsymbol{H}^t(\boldsymbol{x}^t - \bar{\boldsymbol{\mu}}^t) \tag{A.38}$$

The implementation of the extended Kalman filter is quite similar to the linear one as shown in Algorithm 9.

Algorithm 9: Extended Kalman filter algorithm

Input: $\boldsymbol{\mu}^{t-1}$, $\boldsymbol{\Sigma}^{t-1}$, $\boldsymbol{u}^t$ and $\boldsymbol{z}^t$
Output: $\boldsymbol{\mu}^t$ and $\boldsymbol{\Sigma}^t$

/*Prediction stage*/
1 $\bar{\boldsymbol{\mu}}^t = g(\boldsymbol{u}^t, \boldsymbol{\mu}^{t-1})$;
2 $\bar{\boldsymbol{\Sigma}}^t = \boldsymbol{G}^t\boldsymbol{\Sigma}^{t-1}(\boldsymbol{G}^t)^T + \boldsymbol{R}^t$;

/*Correction stage*/
3 $\boldsymbol{S} = \boldsymbol{H}^t\bar{\boldsymbol{\Sigma}}^t(\boldsymbol{H}^t)^T + \boldsymbol{Q}^t$;
4 $\boldsymbol{K}^t = \bar{\boldsymbol{\Sigma}}^t(\boldsymbol{H}^t)^T\boldsymbol{S}^{-1}$;
5 $\boldsymbol{\mu}^t = \bar{\boldsymbol{\mu}}^t + \boldsymbol{K}^t(\boldsymbol{z}^t - h(\bar{\boldsymbol{\mu}}^t))$;
6 $\boldsymbol{\Sigma}^t = (\boldsymbol{I} - \boldsymbol{K}^t\boldsymbol{H}^t)\bar{\boldsymbol{\Sigma}}^t$;

Other implementations

In addition to the Kalman filter and the extended Kalman filter, there are other implementations of the Gaussian filter that in general are lower or equally efficient to Kalman filters (KF and EKF). Some of this implementations are based on different linearization methods like the moments matching (assumed density filter - ADF), which calculates the linearization in a way that preserves the true mean and covariance of the posterior distribution. The problem of this filter is that, despite the approximation to the true mean and covariance of the posterior distribution, it make some assumptions on the moments equations which is inconsistent in general and a bad selection of the number of moments equation may lead to that inconsistencies as explained in [13].

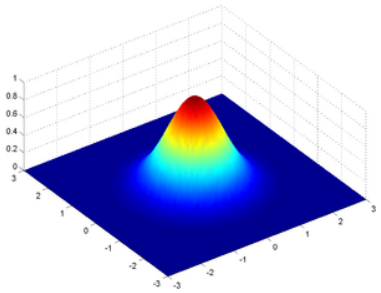
Another common implementation of the Gaussian filter is known as the Unscented Kalman Filter (UKF). This algorithm origins in the unscented transform

as a method of linearization. The method uses a set of so called sigma points in order to sample deterministically the space around the mean of the current Gaussian distribution and assigning weights to these sigma points to predict the new mean and covariance for each new motion vector received. Then the non-linear function g is then applied to each sigma point checking how this function g changes the shape of the Gaussian. To compute the predicted observation, the same process of linearization from sigma points is applied to the observation model. The method has the advantage of not requiring to calculate the Jacobians of the non-linear functions g and h, and use to provide better results than EKF. The UKF propagates the PDF in a simple and effective way and it is accurate up to second order in estimating mean and covariance [62]. In addition the computational complexity is virtually the same of the extended Kalman filter. The draw back of this method is that of the Gaussian filters, they only model uni-modal distributions and hence doesn't fit with other non-Gaussian distributions.

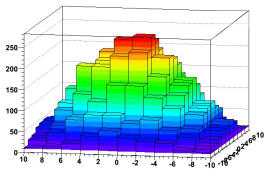
Extended Information filters (EIF) presented in [73] are an equivalent implementation of EKF [2] which employ the canonical parametrization of a Gaussian instead the moments representation used by Kalman filters. The canonical representation is composed by an information matrix $\mathbf{\Omega} = \mathbf{\Sigma}^{-1}$ and an information vector $\boldsymbol{\xi} = \mathbf{\Sigma}^{-1}\boldsymbol{\mu}$. One of the main advantages of this filter is that a complete uncertainty of the initial state can be represented by simply setting the values of the information filter to zero. Furthermore, the extended information filter tends to be more stable numerically than EKF. However, the EIF needs to recover the mean value of the moments representation to apply the non-linear functions to the current state. Despite, the information filter is inefficient compared to EKF for higher dimensions of the state vector, some solutions have been proposed for the case where the matrix to be inverted in the information filter is sparse.

Finally, to overcome the problem of multi-modal distributions, some authors have proposed a semi-parametric, multi-hypotheses algorithms which can dale with this situations [17, 67]. The multi-hypothesis solution for Kalman filters is known as the Gaussian Mixture Model (GMM). This solution will be presented later as a part of the solution proposed for RO-SLAM.

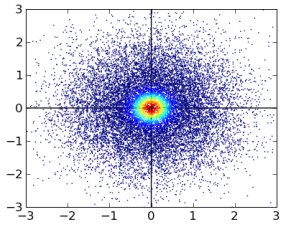
[2] Information filters (IF) are the equivalent implementation of KF for canonical parametrization.



(a) Continuous 2D Gaussian



(b) Grid-based Gaussian



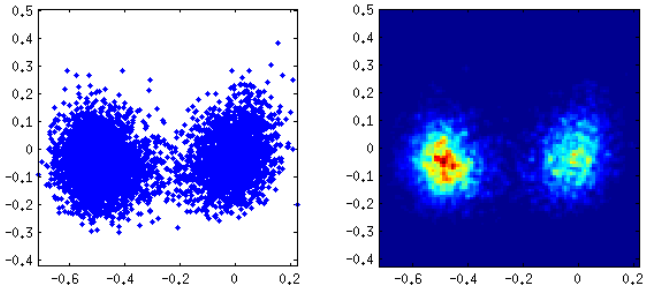
(c) Particle-based Gaussian

Figure A.5: Different representations of a 2D Gaussian distribution.

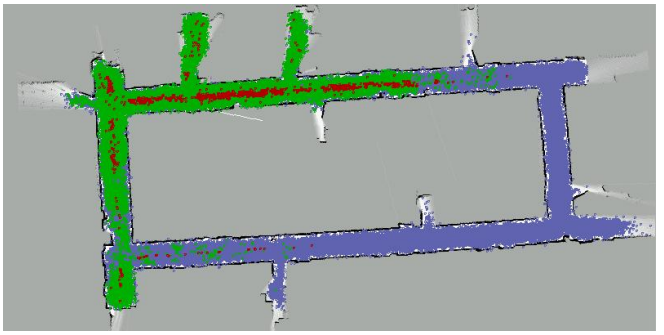
A.3.4 Non-parametric filters

Unlike previous Gaussian filters, non-parametric filters are not based on a function to describe the probability distribution, instead, this kind of filters use a set of ordered or random samples which cover completely or partially the complete state space by sampling it. Hence the number of parameters of this filters is not fixed, since the algorithm developed can use the number of samples necessary to solve each particular problem. Moreover, the efficiency of these algorithms rely on the number of parameters/samples used, so that the larger the number of parameters used, the greater the precision obtained and the greater the amount of computational resources required.

In Figure A.5, an example of a normal distribution is shown with three different representations, the first one (Figure A.5a) uses the classical continuous 2D Gaussian bell (A.2), which is a probability distribution used in Gaussian filters. The second (Figure A.5b) and third (Figure A.5c) representations are sample-based representations typical from non-parametric filters.



(a) Multi-modal distribution



(b) Non-Gaussian distribution

Figure A.6: Typical probability distributions in non-parametric filters

In the examples shown in Figure A.5, despite the probabilistic distribution represented by these sample-based representations is based on a Gaussian distribution, one of the main advantages of non-parametric representations is its ability to represent multi-modal probability distributions, as is the case of Figure A.6a, and other non-Gaussian distributions like in Figure A.6b.

A common improvement of non-parametric filters is the use of adaptive techniques. Adaptive techniques in non-parametric filters try to adapt the number of required parameters in the initialization phase or even during the execution of

the algorithm as the distribution of samples concentrates to certain area of the state space in order to reduce the computational resources of the method.

The Particle Filter - PF

As stated in [78], Sequential Monte Carlo methods (also known as particle filters) are used for filtering and smoothing in general state-space models. These methods are based on importance sampling where each sample is called a particle. This method, as other non-parametric methods, approximate the posterior $bel(\boldsymbol{x}^t) = p(\boldsymbol{x}^t|\boldsymbol{u}^t, \boldsymbol{z}^t)$ by a finite number of particles. The main difference with respect other non-parametric filters is that, in particle filters, particles are generated randomly from the posterior $bel(\boldsymbol{x}^t)$. But, as they are non-parametric, the amount of distributions that can represent is greater than the ones represented by Gaussian filters (even for the case of multi-hypotheses Gaussian filters). As it is a sample-based filter, they are able to model non-linear transformations without having to approximate the non-linear model with linearization techniques as is the case of most Gaussian filters explained above.

Particles are a set P^t of concrete instantiations of the state at a particular instant t, and they are denoted as:

$$P^t = \boldsymbol{p}_1^t, \boldsymbol{p}_2^t, \cdots, \boldsymbol{p}_N^t \tag{A.39}$$

Where N is the number of particles which use to be a larger than 500. With this representation, particles $\boldsymbol{p}_n^t$ can be seen as hypotheses of the state of the true world at time t. The likelihood of a particle is represented by a weight ω_n^t associated to each particle $\boldsymbol{p}_n^t$, then they must sum up to 1 (i.e. $\sum_{n=1}^{N} w_n^t) = 1$. The weight for a state hypotheses $\boldsymbol{x}^t$ to be included in the particle set P^t should be proportional to the posterior $bel(\boldsymbol{x}^t)$:

$$\boldsymbol{p}_n^t \sim p(\boldsymbol{x}^t|\boldsymbol{z}^t, \boldsymbol{u}^t) \tag{A.40}$$

Hence, when the density of hypotheses $\boldsymbol{p}_n^t$ in a region of the state space is very high, then the real state must fall into this region.

The first particle filter implementation [50], which is based on the Sequential Importance Resampling method (SIR), is presented in Algorithm 10. In this algorithm the initialization stage has been omitted. The initialization stage consist on initiate N particles with a prior distribution which is assumed to be similar to the real one.

Algorithm 10: General particle filter algorithm

Input: P^{t-1}, $\boldsymbol{u}^t$ and $\boldsymbol{z}^t$
Output: P^t

/*Prediction stage*/
1 **for all** $\bar{\boldsymbol{p}}_n^t$ ***in*** P^{t-1} **do**
2 Draw $\boldsymbol{p}_n^t \sim p(\boldsymbol{p}^t|\boldsymbol{u}^t, \boldsymbol{p}_n^{t-1})$;
/*Importance factor*/
3 $\bar{\omega^t}_n = p(\boldsymbol{z}^t|\bar{\boldsymbol{p}}_n^t)$;

/*Normalize weights*/
4 $\omega_n^t = \bar{\omega^t}_n / \sum_{i=1}^N \bar{\omega^t}_i$

/*Resampling*/
5 **for** $i = 1 \cdots N$ **do**
6 Draw in P_t according to $p(\boldsymbol{p}_i^t = \bar{\boldsymbol{p}}_j^t) = w_j^t$ for $j = 1 \cdots N$;

The first part of the algorithm is the prediction stage, and is aimed to draw a new set of particles $\bar{P}^t$ of the same size than particle set P^{t-1}, according to the probability $p(\boldsymbol{p}^t|\boldsymbol{u}^t, \boldsymbol{p}_n^{t-1})$. This prediction stage consist mainly on the application of the motion model to each particle of the set P^{t-1}, using the new motion vector $\boldsymbol{u}^t$. The correction stage is divided into two parts, the calculation of the importance factor and the step of importance sampling or more commonly known as resampling step [3].

The first step of the correction stage, known as the calculation of the importance factor, is mainly focused on the update of weights ω_n^t of each particle with the probability $p(\boldsymbol{z}^t|\bar{\boldsymbol{p}}_n^t)$. Then, the new weights must be normalized in order to apply the importance sampling step. Once the predicted belief $\bar{bel}(\boldsymbol{x}^t)$ and the weights are updated according to the observation model of the measurements, the resampling step calculates the posterior $bel(\boldsymbol{x}^t) = \eta p(\boldsymbol{z}^t|\bar{\boldsymbol{p}}_n^t)\bar{bel}(\boldsymbol{x}^t)$ by drawing a new set of random particles P^t with size N. The basic sequential importance resampling (SIR) method consist on choosing M random numbers and selecting those particles which correspond to these random numbers. The resulting particle set usually posses many duplicates since the particles are drawn with replacement, as line 7 of the Algorithm 10 shows. Different implementations of the resampling

[3]The resampling step is the most important step of the particle filter algorithm.

step are presented in [92] which improve the variance of the weights making the filter more robust to different situations.

Finally, at the end of each iteration, the expected state can be calculated as follows:

$$\hat{\boldsymbol{x}}^t = E[bel(\boldsymbol{x}^t)] = E[P^t] = \sum_{n=1} N\omega_n^t \boldsymbol{p}_n^t \tag{A.41}$$

Other implementations

In addition to the particle filter, a set of different approaches can be found in the literature. One of these algorithms is the histogram filter, this grid-based approach decomposes the state space into finitely many regions and represent the cumulative posterior for each region by a single probability value. When applied to finite states spaces, the filter is known as discrete Bayes filter, when applied to continuous state spaces the algorithm is called histogram filter.

An example of discrete Bayes filter, is the occupancy grid map algorithm [31]. This algorithm decompose the finite state into regions which values in a binary domain. When the state does not change its state over time, there is another algorithm called binary Bayes filter with static state [92].

The histogram algorithm is a continuous state estimator which, as described above, decompose the state space into a finite set of regions. Hence the accuracy of this method depends on the granularity employed to make the decomposition of the state. For this reason, multiple decomposition methods are proposed in the literature [92], where one of the most adaptive and efficient methods is known as the density tree decomposition. The advantage of dynamic decomposition methods such as the density tree decomposition is that they are able to achieve a higher approximation quality with the same number of regions used in a static decomposition method. In histogram filters, each cell of the grid has associated a uniform probability distribution, similar to the weight of particles in the particle filter, so that each state value has a probability $p(\boldsymbol{x}^t) = \frac{p_k^t}{|\boldsymbol{x}_k^t|}$, where p_k^t is the probability of a region of the histogram and $|\boldsymbol{x}_k^t|$ is the volume of the region. An example of a Gaussian distribution modeled as an histogram filter with static decomposition is depicted in Figure A.5b.

A.4 Summary and conclusions

Probabilistic robotics offers a different solution from classical perception techniques which are based on methods to extract the optimal solution from a set of observations and motion commands. As the dynamic and observation models employed are not exact, probabilistic robotics treat robotics systems as stochastic processes where motion and observation information are modeled with an additive noise which usually follow a zero mean normal distribution. The algorithms of a probabilistic robotics are mainly Bayes estimators based on the Bayes rule. The basis of Bayes rule as well as other basic probabilistic terms and rules were introduced in this chapter. Another important term included in this chapter was the Markov assumption. The Markov assumption is a characteristic by which the current state is considered complete, i.e. other past events and states are summarized in the current state and hence future states can be estimated from this state. Process which follow this assumption are known as Markov chain processes.

The basis of probabilistic theory, and specially the Bayes rule are the main tools employed in Bayes filters as shown in this chapter. The chapter described a generic Bayes filter from which all probabilistic algorithms considered in this work are based. This generic algorithm is composed by two stages, which are the prediction stage and the correction stage. The prediction stage computes a prior distribution of the state belief $\bar{bel}(\boldsymbol{x}^t) = p(\boldsymbol{x}^t|\boldsymbol{u}^t, \boldsymbol{x}^{t-1})$, employing the motion information received $\boldsymbol{u}^t$ and the previous state $\boldsymbol{x}^{t-1}$. On the other hand, the correction stage is aimed to incorporate the measurements information at time t into the filter from the prior distribution so that the posterior is $bel(\boldsymbol{x}^t) = p(\boldsymbol{z}^t|\boldsymbol{x}^t)\bar{bel}(\boldsymbol{x}^t)$.

Based on this generic Bayes filter, the chapter introduced some Bayesian filters, divided into Gaussian filters and Non-parametric filters. As commented in this chapter, Gaussian filters are nowadays one of the most implemented approaches because of the intrinsic nature of motion and measurement models, which use to follow a Gaussian distribution. The Kalman Filter (KF) for linear models, and its extended version (EKF) for non-linear models, were described as an example of Gaussian filters. These implementations has the main advantage of being an efficient Bayesian filter compared with other non-parametric filters which doesn't scale very well with the size of the state vector. Other Gaussian filters are also mentioned without giving many details.

Although Gaussian filters usually fit well with most observation and dynamic

models, there are other models which doesn't follow a Gaussian distribution and, instead, they follow a multi-modal distribution which might be modeled with a non-parametric filter. For non-Gaussian distribution and other multi-modal Gaussian distributions a particle filter is described in this chapter as an example of the most used non-parametric filter. This filter adapts better to these models and might perform better than other solutions when employing a high density of particles. The random characteristic of this particle filter make it more adaptive than other non-parametric distributions but, at the same time, this randomness might cause some issues related with the variance of the distribution in some situations as described in [92].

In this dissertation, two approximations are used, one of them uses a particle filter to model the multi-modality of the range-only observation model, and the other is based on a semi-parametric model called Gaussian Mixture Models which offers a multi-hypothesis solution that can be integrated in Gaussian filters like the Extended Kalman Filter.

Appendix B

Aerial Robot Datasets using Range-only Sensors

B.1 Introduction

This appendix presents different datasets recorded and used along this thesis. They have been published at *http://grvc.us.es/staff/caba/roslam*. Although, these datasets are mainly composed by range-only observations between an aerial platform and several static range-only sensors, they also include other complementary sensor measurements. They are divided in two sets: indoor and outdoor scenarios. The dataset is of particular interest for aerial robots localization, network localization and Simultaneous Localization and Mapping (SLAM) applications using range-only observations when GPS is not available. Some of the datasets are inspired in the application of range-only technology to aerial manipulation and inspection scenarios of the ARCAS European project.

This appendix describes 6 different datasets recorded using the same aerial robot and sensors under different conditions and applications. Indoor datasets are mainly composed by range-only and altimeter measurements recorded at FADA-CATEC indoor testbed. The outdoor experiments include additional measurements from a monocular camera, used for markers detection and localization or for terrain classification for applications like flat landing area recognition.

To the best of our knowledge, this is the first open dataset focussed on range-only sensors in 3D applications, most of the range-only datasets in the literature are deal with ground robots for 2D applications. Thus, for example, one of the most popular datasets is found in [25] for 2D applications, this dataset is gathered in an outdoor field, free of obstacles and other occlusions, thus presenting raw range data uninhibited by multi-path and other radio interferences. The range-only data in the datasets presented here comprises of data from one of two distinctly different radio-based ranging systems. The first is an radio frequency (RF) based system that measures the time delay of a message sent between low-cost, low-

power, RFID tags placed in the environment and a moving transponder to compute the range. The second system is also a radio based system that utilizes ultra-wide band signals and measures the time delay of arrival between two homogeneous nodes to compute the range between the node pair. This dataset is easy to use with any RO-SLAM implementation as it is published as raw plain text files and with well-known Matlab format files. Another interesting dataset is found in [7], the dataset includes range-only measurements from radio-based range sensors and is aimed for 2D and 3D mapping with a ground robot. The dataset uses a raw format for MRPT framework developed at the University of Malaga. For WiFi-based range-only measurements, there is also a dataset available in [55]. This dataset is also published using common Matlab format, so that it can be easily used with any RO-SLAM algorithm. Finally, other datasets using RSS or WiFi technology have been found in CRAWDAD web page [4, 79, 81, 84, 102].

However, most of these datasets are aimed for ground robot applications, and none aerial robot dataset as been found using range-only sensors. The datasets presented in this appendix have been recorded using well-known rosbag tool from ROS framework and can be easily exported into plain text format using the same tool. To the best of our knowledge, is the first aerial robot dataset using range-only sensors. This datasets have been published at [41] for its use in the research community.

B.2 Platform description

During the experiments in which these datasets were recorded, a Pelican quadcopter from Asctec was used as the main aerial robot platform (see Figure B.1). The robot was equipped with a 1.6 GHz Intel Atom board and with 1GB DDR2 RAM. The atom board was used to run all sensor drivers and to record all information with rosbag[1] tool from ROS[2] framework. The ROS distribution used to register these datasets was ROS Fuerte using an Ubuntu 12.04LTS Linux distribution. In order to play back one of the datasets use *rosbag play <experiment>.bag* after installing ROS fuerte or other compatible ROS distribution.

All datasets include the relative frame transforms for sensors robot platform using the standard package *tf* [3] from ROS distribution. Different set of sensors

[1] http://wiki.ros.org/rosbag

[2] http://wiki.ros.org/

[3] http://wiki.ros.org/tf

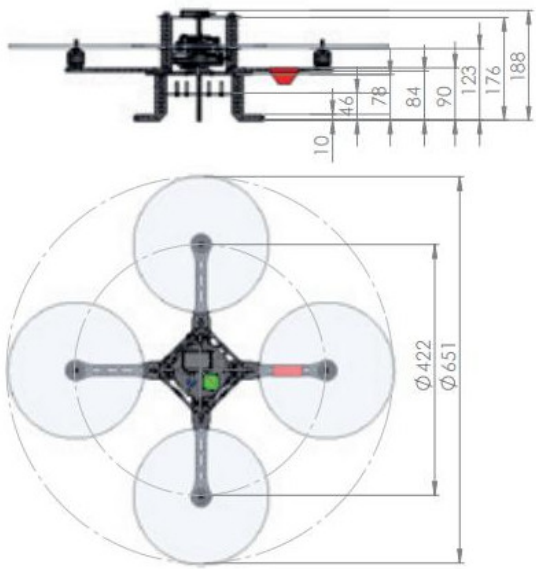


Figure B.1: Technical data of the Pelican quad-copter used during experiments from Ascending Technologies data sheet (sizes in millimeters).

and configurations were used between indoor and outdoor experiments.

The range-only sensor employed in all datasets have been manufactured by Nanotron company. The range-only sensor model employed from this company was the development board which uses the nanoPAN 5375 chip (see Figure B.2). The technical specifications of this range-only sensor are:

- ATMega 1284P microcontroller at 20MHz for range computation and data packages processing.
- Radio transceiver 2.4 GHz ISM band. Up to 20dB transmission power.
- Ranging accuracy of 2m indoors and 1m outdoors.
- Ranging frequency: 80Hz.

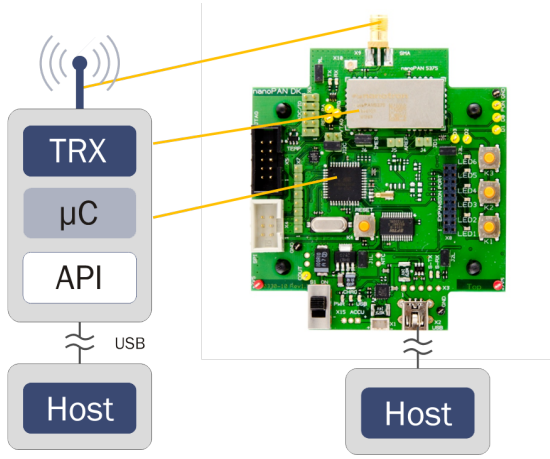


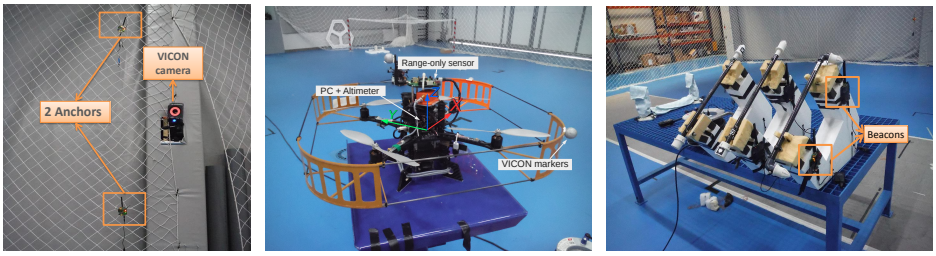
Figure B.2: Range-only sensors employed during experiments: nanoPAN 5375 DB. The figure shows the essential parts of the nanoPAN 5375 radio beacon architecture. The upper layer represents the hardware of this device. The API layer represents the drivers provided by the manufacturer to send ranging commands with other devices, and other auxiliary methods. The last layer represents the host controller, i.e. the computer in which range measurements are processed.

- 128KB flash memory for programs and retrieved data.
- Distance measurements computed with SDS-TWR technique without needing any clock synchronization between nodes.

As part of the experiments carried out to register these datasets, a driver wa developed (ROS package *nanotron-swarm*[4]) for the first version of the Nanotron swarm API using the serial port communication interface with the sensor. The driver uses ROS middle-ware for data distribution and is available for different distributions of ROS. The driver has been published as part of a ROS stack called *rangeonly_driver*[5]. The version of the driver used for these experiments is the one tagged as *rosbuild* in github repository *https://github.com/felramfab/rangeonly_driver*.

The datasets also include, for both environments, altitude measurements from the barometric altimeter connected to the Atom board of the Pelican quad-copter.

[4]http://wiki.ros.org/nanotron_swarm
[5]http://wiki.ros.org/rangeonly_driver



(a) (b) (c)

Figure B.3: Setup used for indoor real experiments: (a) anchors used for localization and VICON system used for groundtruth (b) Pelican aerial vehicle from Asctec (c) 3 bars with 6 embedded beacons.

Figure B.3 shows the configuration of sensors used for the aerial robot during indoor experiments in FADA-CATEC indoor testbed. In this case the robot does not include any additional sensor apart from that already described above. However, the platform include some markers (see Figure B.3b) used to get the ground-truth of the elements (aerial robot, bars and anchors) involved in the experiment using a VICON motion capture system[6] (see Figure B.3a). For indoor environments, some anchors were placed in the bounds of the testbed for localization applications. Other sensors were also placed throughout the working area but at different positions for each dataset with the purpose to use them as landmarks (or beacons) of the RO-SLAM problem. In one of the experiments for indoor environments, beacons are placed as if they were embedded in the bars of a structure to be constructed for aerial manipulation applications (see ARCAS project in Chapter 1).

Figure B.4 shows the setup used for outdoor experiments. In this case, the aerial robot included a monocular camera in its based pointing downwards for the detection of some visual markers placed on the floor or for terrain classification. The platform also includes a GPS antenna connected to a base antenna used as part of a differential GPS system used to compute the ground-truth of the aerial robot. In these experiments, a set of anchors is also used for localization purpose around the flying area of experiments. A set of landmarks is also placed throughout the flying area to be used as landmarks in RO-SLAM.

[6] http://www.vicon.com/

(a) (b) (c)

Figure B.4: Setup used for outdoor real experiments: (a) hangar of aeromodelism club SAETA (Seville, Spain) with several landmarks (or beacons) and base differential GPS antenna (b) anchors used for localization (c) Pelican aerial vehicle from Ascending Technologies.

B.3 Data types

This section details the fields of the data types used in the data sets described in this appendix.

- *rangeonly_msgs::P2PRange*[7]: This data type is used to represent range-only measurements. This ROS data type has been developed as part of this thesis and has been published in ROS site as part of the *rangeonly_driver*[8] stack in ROS package *rangeonly_msgs*[9]. The fields of this data type are:
  - *header*: Common ROS header including message time-stamp, frame identifier of the sensor and other meta-data.
  - *radiation_type*: The type of radiation used by the sensor (inherit field from ROS *sensor_msgs::Range* for reflection-based range sensors).
  - *emitter_id*: Integer which identifies the emitter device unique identifier.

[7] https://github.com/felramfab/rangeonly_driver/blob/indigo-devel/rangeonly_msgs/msg/P2PRange.msg

[8] http://wiki.ros.org/rangeonly_driver

[9] http://wiki.ros.org/rangeonly_msgs

  - *emitter_type*: This field defines whether the emitter sensor is an ANCHOR a BEACON (here called landmark) or a BASE (host sensor connected to computer) sensor.
  - *receiver_id*: Integer which identifies the receiver device unique identifier.
  - *receiver_type*: This field defines whether the receiver sensor is an ANCHOR a BEACON (here called landmark) or a BASE (host sensor connected to computer) sensor.
  - *range*: Double which represents the distance measured from emitter to receiver device in meters.
  - *variance*: This field represents the variance of the range measurement.

- *mav_msgs::Height*[10]: This data type is used the barometric altimeter included as part of the Ascending Technologies Pelican quad-copter. The fields of this data type are:
  - *header*: Common ROS header including message time-stamp, frame identifier of the sensor and other meta-data.
  - *height*: The altitude measurement in meters.
  - *height_variance*: The altitude measurement variance.
  - *climb*: The vertical speed measurement in meters per second.
  - *climb_variance*: The vertical speed measurement variance.

- *geometry_msgs::PoseStamped*[11]: This data type is used to record the ground-truth of the aerial robot. The fields of this data type are:
  - *header*: Common ROS header including message time-stamp, frame identifier of the sensor and other meta-data.
  - *pose.position*: The 3D position of the robot in meters.
  - *pose.orientation*: The orientation of the aerial robot in quaternion format.

[10] https://github.com/ccny-ros-pkg/mav_tools/blob/master/mav_msgs/msg/Height.msg
[11] http://docs.ros.org/jade/api/geometry_msgs/html/msg/PoseStamped.html

- *sensor_msgs/CompressedImage*[12]: This data type is used to record the images of the experiments in compressed format. The fields of this data type are:
  - *header*: Common ROS header including message time-stamp, frame identifier of the camera and other meta-data.
  - *format*: The image format (jpeg, png, etc).
  - *data*: The compressed image buffer.
- *sensor_msgs/CameraInfo*[13]: This data type is used to record information about the camera like calibration parameters. The main fields of this data type are:
  - *header*: Common ROS header including message time-stamp, frame identifier of the camera and other meta-data.
  - *height*: The height of the image in pixels.
  - *width*: The width of the image in pixels.
  - *distortion_model*: The distortion model used by the camera.
  - *D*: The parameters of the distortion model.
  - *K*: The intrinsic camera parameters.
  - *R*: This field is not used in these experiments (used for stereo cameras).
  - *P*: The intrinsic parameters of the rectified image.
- *gps_common/GPSFix*[14]: This data type is used to record information from differential GPS sensor using a common data type used by ROS community. The main fields of this data type are:
  - *header*: Common ROS header including message time-stamp, frame identifier of the camera and other meta-data.
  - *status*: The state of GPS signal. This data type contains information like the number of satellites visible and used, the SNR of the signal or the measurement status (fix,no fix, etc).

[12]http://docs.ros.org/jade/api/sensor_msgs/html/msg/CompressedImage.html
[13]http://docs.ros.org/jade/api/sensor_msgs/html/msg/CameraImage.html
[14]http://docs.ros.org/hydro/api/gps_common/html/msg/GPSFix.html

  - *latitude*: The geographical latitude of the vehicle in degrees.
  - *longitude*: The geographical longitude of the vehicle in degrees.
  - *altitude*: The altitude of the vehicle with respect sea level in meters.
  - *track*: The vehicle direction with respect north in degrees.
  - *speed*: The ground vehicle speed in meters per second.
  - *climb*: The vertical speed of the vehicle in meters per second.

- *sensor_msgs/NavSatFix*[15]: This data type is used to record information from differential GPS sensor using ROS standard data type. The main fields of this data type are:

  - *header*: Common ROS header including message time-stamp, frame identifier of the camera and other meta-data.
  - *status*: The state of GPS signal. This data type contains information like the number of satellites visible and used, the SNR of the signal or the measurement status (fix,no fix, etc).
  - *latitude*: The geographical latitude of the vehicle in degrees.
  - *longitude*: The geographical longitude of the vehicle in degrees.
  - *altitude*: The altitude of the vehicle with respect sea level in meters.

- *tf2_msgs/TFMessage*[16] *or tf_msgs/tfMessage*[17]: This data type is used to record the relative frame transformations. This data type contains a list of elements of type *geometry_msgs::TransformStamped*[18]. The fields of each transform in the list are:

  - *header*: Common ROS header including message time-stamp, parent frame identifier of the sensor and other meta-data.
  - *child_frame_id*: The child frame identifier, i.e. the frame identifier to which points are transformed from the frame id given in *header.frame_id* field.

[15] http://docs.ros.org/jade/api/sensor_msgs/html/msg/NavSatFix.html
[16] http://docs.ros.org/jade/api/tf2_msgs/html/msg/TFMessage.html
[17] http://docs.ros.org/jade/api/tf/html/msg/tfMessage.html
[18] http://docs.ros.org/jade/api/geometry_msgs/html/msg/TransformStamped.html

Figure B.5: FADA-CATEC indoor testbed used during experiments in Seville (Spain).

- *transform.translation*: The translation from frame with identifier *header.frame_id* to frame *child_frame_id*.
- *transform.rotation*: The rotation in quaternions from frame wit identifier *header.frame_id* to frame with identifier *child_frame_id*.

B.4 Indoor datasets description

These datasets are divided in 3 experiments recorded in three different ROS bags at FADA-CATEC indoor testbed. The main purpose of these experiments is to evaluate the results of using range-only sensors in the absence of GPS signal. As introduced before, in these experiments only an altimeter and range-only sensors are used for estimations. The VICON motion capture system is used to get the ground-truth of the robot and landmarks with millimeters of precision.

These experiments were carried out at the indoor testbed of FADA-CATEC in Seville (Spain). The testbed (see Figure B.5) size is about $16 \times 15 \times 6$ meters with more than 10 UAVs available for experiments.

During the experiments 22 range-only sensors were deployed, where 16 of them where attached to the fence of the testbed at different altitudes with identifier from 6 to 21, and the other 6 with identifiers from 1, 2, 4, 22, 23 and 24 were where deployed at different testbed positions depending on the dataset. The aerial robot was connected to the base range-only sensor with identifier 25. The base device was in charge of sending the range requests to all anchors and beacons, receive the range measurements and send them to the robot computer through the serial interface. It should be notice that sensors located in the fence of the testbed might be used as anchors for localization or as landmarks for mapping, however their position might not be suitable for their tracking from the robot trajectory.

Due to multi-path, reflection, absorption and other effects caused by walls, floor and other reflective materials of the indoor testbed, the dataset might contain some outliers or biased range measurements. This wrong measurements might be

used to evaluate some algorithms like outliers detectors and other bias estimation methods like the one described in chapter 4 to make RO-SLAM approaches more robust against these kind of effects.

B.4.1 Data details

The bag files per indoor dataset contain the following topics or source of information:

- */mav/pressure_height_filtered*: This topic is used to send barometric altimeter measurements from Pelican integrated altimeter with data type *mav_msgs::Height*.
- */nanotron/range*: This topic is used to send range-only measurements received from serial port connected to base range-only sensor using data type *nanotron_swarm::P2PRange*.
- */vicon_proxy_receiver/pose*: This topic is used to send the estimated ground-truth of the robot from the VICON tracking system using data type *geometry_msgs::PoseStamped*.
- */vicon_proxy_receiver/odom*: This topic is also used to send the estimated ground-truth of the robot from the VICON tracking system using data type *nav_msgs::Odometry*.
- */tf*: This topic is used to send the relative transformations between system references used in the experiments using data type *tf2_msgs::TFMessage*.

In these experiments, the world system reference used to get the ground-truth of the aerial robot and range-only sensors is placed in the right corner close to the operators room of the indoor tesbed and with zero altitude corresponding to the floor of the testbed. Other system references used in these experiments for sensors and aerial robot are:

- **Aerial robot frame:** with ROS frame identifier: */base_link*.
- **Range-only frame:** with ROS frame identifier: */radio_range*.
- **Barometric altimeter frame:** with ROS frame identifier: */imu*.

Additionally, these datasets contains a pair of files *groundtruth_dataset.mat* and *node_poses.mat*. The first file contains the ground truth of the robot position extracted from the bag file (topic /vicon_proxy_receiver/pose) using *rosbag* library. The second file contains the real position and unique identifiers of the range-only sensors.

B.4.2 Experiment 1: Aerial manipulation

This experiment is aimed for the vehicle localization and the mapping of some beacons attached to a set of structure bars to be manipulated. Each bar was attached to two beacons, each of them identifying the borders of the bars as shown in Figure B.3c.

Figure B.6 shows the 3D, 2D and altitude trajectory followed by the aerial robot during the experiment. First of all the robot takes off and flies around the area to get a first estimation of bars position, later the robot moves closer to bars position in order to refine the position of the bars and to simulate the position from which the manipulator could star operating. A video of this experiment has been published at *https://youtu.be/KNRebI31nrg*.

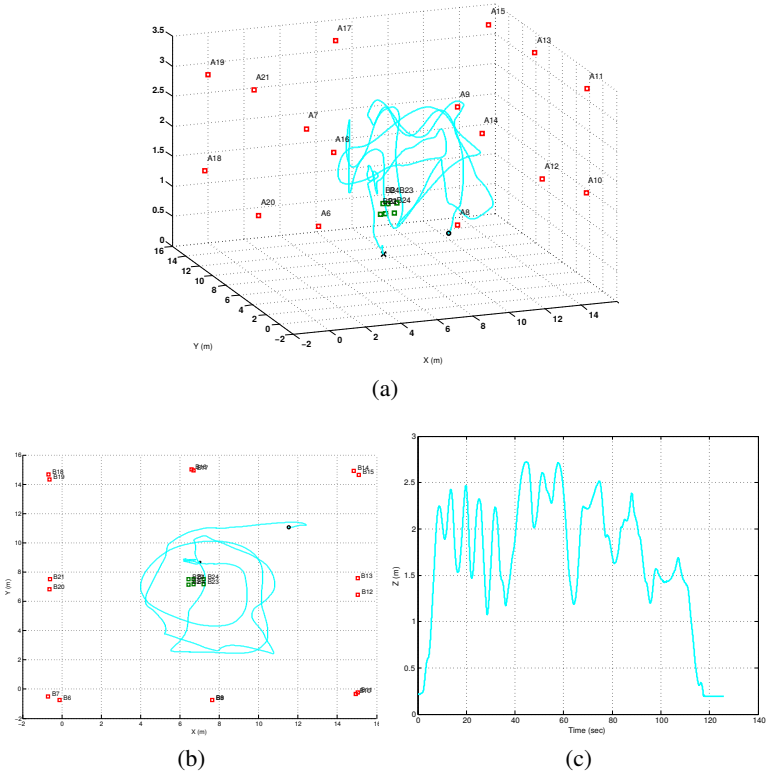
B.4.3 Experiment 2: Circular exploration

This experiment is aimed for the vehicle localization and the mapping of some beacons attached to a set of pipes to be inspected. As shown in Figure B.7, in this experiment, beacons are placed on top of pipes and near the floor. In this experiment beacon 22 is not used.

Figure B.8 shows the 3D, 2D and altitude trajectory followed by the aerial robot during the experiment. In this case the robot starts flting with a circular trajectory around the environment and later it approaches to one of the pipes. A video of this experiment has been published at *https://youtu.be/a6LdY6_IKOY*.

B.4.4 Experiment 3: Zig-zag exploration

This experiment is aimed for the vehicle localization and the mapping of some beacons attached to a set of pipes to be inspected. This experiment uses the same setup of previous experiment with circular trajectory, but using a different setup on the position of beacons.This experiments includes beacon 22 in the dataset.



(a) (b) (c)

Figure B.6: Trajectory of the robot during aerial manipulation experiment. Initial position represented as a black circle and final position represented as a black cross. Green squares represent the real position of beacons whereas red squares represent the position of anchors: (a) 3D view, (b) horizontal view and (c) vertical view

Figure B.9 shows the 3D, 2D and altitude trajectory followed by the aerial robot during the experiment. In this case the robot starts flying with a zig-zag trajectory around the environment and later it approaches to one of the pipes. A video of this experiment has been published at *https://youtu.be/Rk3hpnXynZQ*.

B.5 Outdoor datasets description

These datasets are also divided in 3 experiments which main purpose is the use of range-only devices as a complementary sensor to visual observations for a

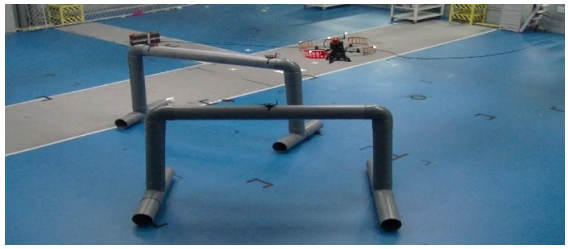


Figure B.7: Setup of the environment used for exploration of mockup of an industrial environment with pipes.

more precise and continuous state estimation. Range-only devices allow to have a first estimation of landmarks when the sensors are not yet in the field of view of the camera. In this case, some of the range-only sensors were placed around the flying area so that they can be used as anchors for robot localization, however these sensors may also be used as landmarks. The monocular camera can also be used to compute a visual odometry of the aerial robot platform during the prediction stage of the filter. The datasets also includes the altimeter information for a better altitude estimation. The differential GPS was mainly introduced in these datasets for ground truth estimation, however, the data can also be used with other purposes.

These experiments were carried out at the aeromodelling club RC SAETA in Seville (Spain) (see Figure B.10). The experiment environment is composed by a takeoff/landing track and is surrounded by vegetation. This combination of paved land and vegetation might be used for landing area recognition algorithms.

During the experiments 18 range-only sensors were deployed around the area between the vegetation, in the fence and along the landing track. The altitude of this range-only sensors is quite similar, only using some elements to place them at different altitudes like chairs or tripods as shown in Figure B.4a.

Furthermore, 33 visual markers where placed around the takeoff area (see Figure B.4a). These markers might be used as features to detect the landing area or to have an increased precision of the robot position with respect range-only sensors precision. These markers were generated using the Augmented Reality library (ArUco http://www.uco.es/investiga/grupos/ava/node/26) developed by the University of Cordoba. The markers are specially designed to be recognized by this library with a precision of about 15cm, other libraries will probably not recognize them.

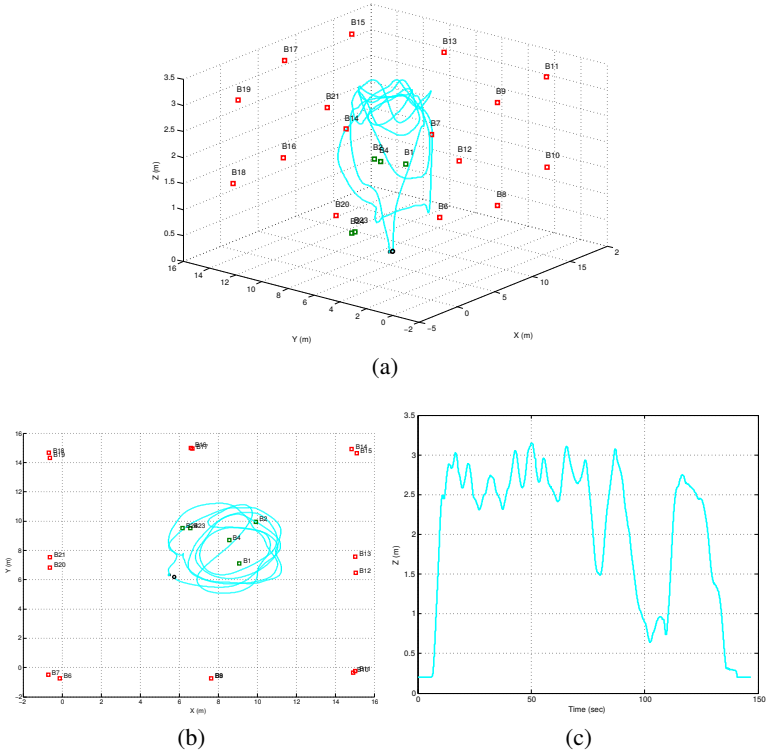
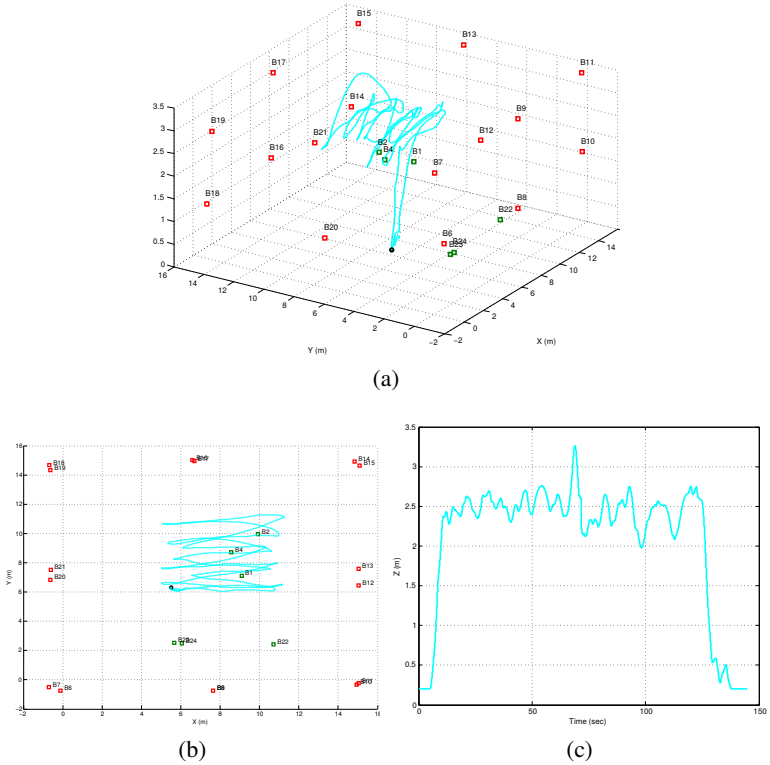


Figure B.8: Trajectory of the robot during aerial exploration of an environments with pipes. Initial position represented as a black circle and final position represented as a black cross. Green squares represent the real position of beacons whereas red squares represent the position of anchors: (a) 3D view, (b) horizontal view and (c) vertical view

The aerial robot was connected to the base range-only sensor with identifier 25 and to a HD webcam Logitech c920 camera. The base range-only device was in charge of sending the range requests to all anchors and beacons, receive the range measurements and send them to the robot computer through the serial interface. The camera was attached to the bottom of the quadcopter using rubber dampers to avoid vibrations on the image. The technical specifications of this camera are:

- Full HD video recording (up to 1920×1080 pixels) in these experiments limited to 600×800 pixels to reduce the jello-effect.



(a)

(b) (c)

Figure B.9: Trajectory of the robot during aerial exploration of an environments with pipes. Initial position represented as a black circle and final position represented as a black cross. Green squares represent the real position of beacons whereas red squares represent the position of anchors: (a) 3D view, (b) horizontal view and (c) vertical view

- Rolling shutter.
- H.264 video compression.
- Automatic low-light correction.

The intrinsic and extrinsic parameters of the camera are included in the */logitech/camera_info* topic of the dataset.

Figure B.10: RC SAETA aeromodelling club used during outdoor experiments in Seville (Spain).

B.5.1 Data details

The bag files per outdoor dataset contain the following topics or source of information:

- */mav/pressure_height_filtered*: This topic is used to send barometric altimeter measurements from Pelican integrated altimeter with data type *mav_msgs::Height*.

- */nanotron/range*: This topic is used to send range-only measurements received from serial port connected to base range-only sensor using data type *nanotron_swarm::P2PRange*.

- */logitech/image_raw/compressed*: This topic is used to send camera images using data type *sensor_msgs::CompressedImage*.

- */logitech/camera_info*: This topic is used to send camera parameters using data type *sensor_msgs::CameraInfo*.

- */pose*: This topic is used to send the estimated ground-truth of the robot from the differential GPS using data type *geometry_msgs::PoseStamped*.

- */rtkekf/gnss_gpsfix_filtered*: This topic is used to send the estimated GPS position using an EKF and the observations from the differential GPS unit using data type *gps_common::GPSFix*.

- */rtkekf/gnss_navsatfix_filtered*: This topic is used to send the estimated GPS position using an EKF and the observations from the differential GPS unit using data type *sensor_messages::NavSatFix*.
- */rtkekf/gnss_gpsfix*: This topic is used to send the raw GPS observations from the differential GPS unit using data type *gps_common::GPSFix*.
- */rtkekf/gnss_navsatfix*: This topic is used to send the raw GPS observations from the differential GPS unit using data type *sensor_messages::NavSatFix*.
- */tf*: This topic is used to send the relative transformations between system references used in the experiments using data type *tf::TFMessage*.

In these experiments, the world system reference used to get the ground-truth of the aerial robot, visual markers and range-only sensors is placed in the center of the takeoff/landing track. Other system references used in these experiments for sensors and aerial robot are:

- **Aerial robot frame:** with ROS frame identifier: */base_link*.
- **Range-only frame:** with ROS frame identifier: */radio_range*.
- **Barometric altimeter frame:** with ROS frame identifier: */imu*.

These datasets also contains a pair of files *groundtruth_dataset.mat* and *node_poses.mat*. The first file contains the ground truth of the robot position extracted from the bag file (topic /vicon_proxy_receiver/pose) using *rosbag* library. The second file contains the real position and unique identifiers of the range-only sensors.

Additionally, these outdoor datasets include a file *marker_poses.mat* which contains the real position of the visual markers, their unique identifiers and their size. As the shape of these markers is squared, their size is given by the side size.

B.5.2 Experiment 1: Landing with range-only and visual measurements

This experiment is aimed at developing algorithms for the precise localization of the robot during the landing maneuver. In this experiments, range-only sensors

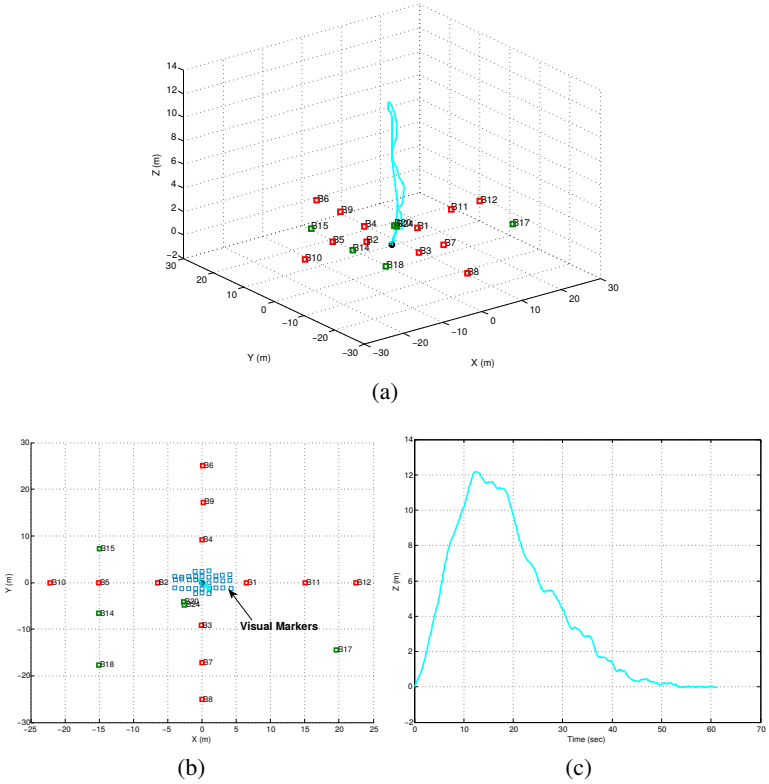


Figure B.11: Trajectory of the robot during landing maneuver. Initial position represented as a black circle and final position represented as a black cross. Green squares represent the real position of beacons whereas red squares represent the position of anchors. The blue squares represent the real position of visual markers: (a) 3D view, (b) horizontal view and (c) vertical view

are advised to be used as anchors, since the trajectory of the robot does not allow a good trilateration of the sensors.

Figure B.11 shows the 3D, 2D and altitude trajectory followed by the aerial robot during the experiment. In this case the robot takes off from the center of the take off track and, after reaching 12 meters of altitude, the robot starts the landing maneuver. A video of this experiment has been published at *https://youtu.be/hJbkWzM7xAg*.

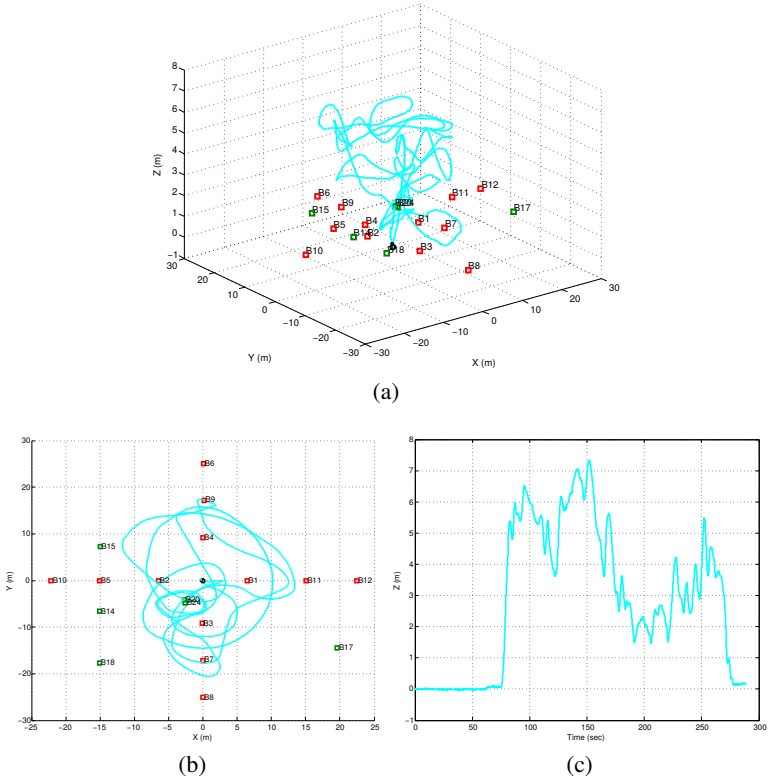
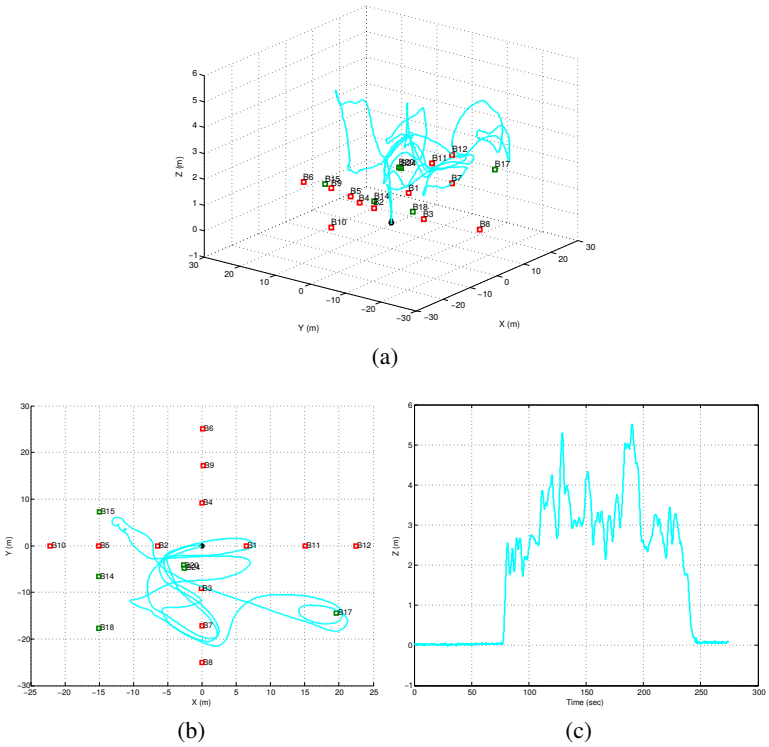


Figure B.12: Trajectory of the robot during range-only sensors mapping. Initial position represented as a black circle and final position represented as a black cross. Green squares represent the real position of beacons whereas red squares represent the position of anchors: (a) 3D view, (b) horizontal view and (c) vertical view

B.5.3 Experiment 2: Aerial trilateration of beacons

This experiment is aimed at developing exploration algorithms for the localization of the aerial robot using local sensors and/or mapping of a set of range-only sensors.

Figure B.12 shows the 3D, 2D and altitude trajectory followed by the aerial robot during the exploration. In this case the robot takes off from the center of the take off track and, after some exploration maneuvers around different range-only sensors, the robot lands again in the same take off point. A video of this experiment has been published at *https://youtu.be/V3garqWyYx0*.



(a)

(b) (c)

Figure B.13: Trajectory of the robot during exploration and landing area recognition. Initial position represented as a black circle and final position represented as a black cross. Green squares represent the real position of beacons whereas red squares represent the position of anchors: (a) 3D view, (b) horizontal view and (c) vertical view

B.5.4 Experiment 3: Aerial mapping and landing area recognition

This experiment is aimed at developing exploration algorithms for the localization of the aerial robot using local sensors and/or mapping of a set of range-only sensors. In this case the robot follows always almost the same altitude with the intention of detecting all possible landing areas.

Figure B.13 shows the 3D, 2D and altitude trajectory followed by the aerial robot during the exploration. In this case the robot takes off from the center of the take off track and, after some exploration maneuvers around the environment, the robot lands again in the same take off point. A video of this experiment has been published at *https://youtu.be/dDDuBdFUdCM*.

References

[1] AHMAD, A., HUANG, S., WANG, J., AND DISSANAYAKE, G. A new state vector and a map joining algorithm for range-only SLAM. In *2012 12th International Conference on Control Automation Robotics Vision (ICARCV)* (2012), pp. 1024–1029.

[2] AHN, H., HUR, H., AND CHOI, W. One-way ranging technique for CSS-based indoor localization. In *Industrial Informatics, 2008. INDIN 2008. 6th IEEE International Conference on* (2008), pp. 1513–1518.

[3] AMOR-MARTINEZ, A., RUIZ, A., MORENO-NOGUER, F., AND SANFELIU, A. On-board real-time pose estimation for UAVs using deformable visual contour registration. In *2014 IEEE International Conference on Robotics and Automation (ICRA)* (May 2014), pp. 2595–2601.

[4] BAUER, KEVIN, W. ANDERSON, ERIC, MCCOY, DAMON, GRUNWALD, DIRK, AND C. SICKER, DOUGLAS. CRAWDAD dataset cu/rssi (v. 2009-05-28), May 2009.

[5] BESSIÈRE, PIERRE, LAUGIER, CHRISTIAN, AND SIEGWART, ROLAND. *Probabilistic Reasoning and Decision Making in Sensory-Motor Systems*, vol. 46 of *1610-7438*. Springer Berlin Heidelberg, 2008.

[6] BISCHOFF, O., WANG, X., HEIDMANN, N., LAUR, R., AND PAUL, S. Implementation of an ultrasonic distance measuring system with kalman filtering in wireless sensor networks for transport logistics. *Procedia Engineering 5* (2010), 196–199.

[7] BLANCO, J. Dataset for bayesian range-only SLAM (RO-SLAM) with SOGs | MRPT, Oct. 2013.

[8] BLANCO, J.-L., FERNANDEZ-MADRIGAL, J.-A., AND GONZALEZ, J. Efficient probabilistic range-only SLAM. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2008. IROS 2008* (Sept. 2008), pp. 1017 –1022.

[9] BLANCO, J.-L., GONZALEZ, J., AND FERNANDEZ-MADRIGAL, J. A pure probabilistic approach to range-only SLAM. In *IEEE International Conference on Robotics and Automation, 2008. ICRA 2008* (May 2008), pp. 1436–1441.

[10] BOOTS, B., AND GORDON, G. J. A spectral learning approach to range-only SLAM. *arXiv:1207.2491* (July 2012).

[11] BOUABDALLAH, S., AND SIEGWART, R. Full control of a quadrotor. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2007. IROS 2007* (2007), pp. 153–158.

[12] BOURGAULT, F., MAKARENKO, A., WILLIAMS, S., GROCHOLSKY, B., AND DURRANT-WHYTE, H. Information based adaptive robotic exploration. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems* (2002), pp. 540–545.

[13] BRIGO, D., HANZON, B., AND GLAND, F. L. On the relationship between assumed density filters and projection filters. In *Discussion Paper TI 7-96-18, Tinbergen Institute* (1996).

[14] CABALLERO, F., MERINO, L., FERRUZ, J., AND OLLERO, A. Vision-based odometry and SLAM for medium and high altitude flying uavs. *J. Intell. Robotics Syst. 54*, 1-3 (Mar. 2009), 137–161.

[15] CABALLERO, F., MERINO, L., GIL, P., MAZA, I., AND OLLERO, A. A probabilistic framework for entire WSN localization using a mobile robot. *Robotics and Autonomous Systems 56*, 10 (Oct. 2008), 798–806.

[16] CABALLERO, F., MERINO, L., MAZA, I., AND OLLERO, A. A particle filtering method for wireless sensor network localization with an aerial robot beacon. In *IEEE International Conference on Robotics and Automation, 2008. ICRA 2008* (May 2008), pp. 596 –601.

[17] CABALLERO, F., MERINO, L., AND OLLERO, A. A general gaussian-mixture approach for range-only mapping using multiple hypotheses. In *IEEE International Conference on Robotics and Automation (ICRA), 2010* (2010), pp. 4404–4409.

[18] CARRILLO-ARCE, L., NERURKAR, E., GORDILLO, J., AND ROUMELIOTIS, S. Decentralized multi-robot cooperative localization using covariance intersection. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Nov. 2013), pp. 1412–1417.

[19] CASTELLANOS, J., NEIRA, J., AND TARDOS, J. Multisensor fusion for simultaneous localization and map building. *IEEE Transactions on Robotics and Automation 17*, 6 (Dec. 2001), 908–914.

[20] CHO, H., JUNG, Y., CHOI, H., JANG, H., SON, S., AND BAEK, Y. Real time locating system for wireless networks using IEEE 802.15.4 radio. In *5th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, 2008. SECON '08* (June 2008), IEEE, pp. 578–580.

[21] COMSA, C.-R., LUO, J., HAIMOVICH, A., AND SCHWARTZ, S. Wireless Localization using Time Difference of Arrival in Narrow-Band Multipath Systems. In *International Symposium on Signals, Circuits and Systems, 2007. ISSCS 2007* (July 2007), vol. 2, pp. 1–4.

[22] COOPER, G. F. The computational complexity of probabilistic inference using bayesian belief networks. *Artificial Intelligence 42*, 2-3 (Mar. 1990), 393–405.

[23] DAVISON, A. J., REID, I. D., MOLTON, N. D., AND STASSE, O. Monoslam: Real-time single camera slam. *IEEE Trans. Pattern Analysis and Machine Intelligence 29* (2007), 2007.

[24] DJUGASH, J. *Geolocation with Range: Robustness, Efficiency and Scalability*. PhD thesis, Carnegie Mellon University - CMU, Nov. 2010.

[25] DJUGASH, J., HAMNER, B., AND ROTH, S. Navigating with ranging radios: Five data sets with ground truth. *Journal of Field Robotics 26*, 9 (Sept. 2009), 689–695.

[26] DJUGASH, J., AND SINGH, S. A robust method of localization and mapping using only range. In *Experimental Robotics*, O. Khatib, V. Kumar, and G. J. Pappas, Eds., no. 54 in Springer Tracts in Advanced Robotics. Springer Berlin Heidelberg, Jan. 2009, pp. 341–351.

[27] DJUGASH, J., AND SINGH, S. Motion-aided network SLAM with range. *The International Journal of Robotics Research 31*, 5 (Apr. 2012), 603 – 625.

[28] DJUGASH, J., AND SINGH, S. Motion-aided network SLAM. In *Experimental Robotics*, O. Khatib, V. Kumar, and G. Sukhatme, Eds., no. 79 in Springer Tracts in Advanced Robotics. Springer Berlin Heidelberg, Jan. 2014, pp. 447–460.

[29] DJUGASH, J., SINGH, S., AND GROCHOLSKY, B. Decentralized mapping of robot-aided sensor networks. In *IEEE International Conference on Robotics and Automation, 2008. ICRA 2008* (May 2008), pp. 583 –589.

[30] DJUGASH, J., SINGH, S., KANTOR, G., AND ZHANG, W. Range-only SLAM for robots operating cooperatively with sensor networks. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006* (2006), pp. 2078–2084.

[31] ELFES, A. Using occupancy grids for mobile robot perception and navigation. *Computer 22*, 6 (1989), 46–57.

[32] FABRESSE, F. R. General purpose range-only sensor data message package for ros framework. `http://wiki.ros.org/rangeonly_msgs`.

[33] FABRESSE, F. R. Range-only driver stack for ros framework. `http://wiki.ros.org/rangeonly_driver`.

[34] FABRESSE, F. R., CABALLERO, F., MAZA, I., AND OLLERO, A. Localización de vehículos aéreos basada en 3D RO-SLAM con inicialización no retardada empleando mezcla de Gaussianas. In *ROBOT 2013: First Iberian Robotics Conference* (Centre for Automation and Robotics (CAR), Madrid, Spain, Nov. 2013).

[35] FABRESSE, F. R., CABALLERO, F., MAZA, I., AND OLLERO, A. Undelayed 3d RO-SLAM based on gaussian-mixture and reduced spherical

parametrization. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Tokyo Big Sight, Tokyo, Japan, Nov. 2013), pp. 1555–1561.

[36] FABRESSE, F. R., CABALLERO, F., MAZA, I., AND OLLERO, A. Localization and mapping for aerial manipulation based on range-only measurements and visual markers. In *2014 IEEE International Conference on Robotics and Automation (ICRA)* (May 2014), pp. 2100–2106.

[37] FABRESSE, F. R., CABALLERO, F., MAZA, I., AND OLLERO, A. Robust range-only SLAM for aerial vehicles. In *2014 International Conference on Unmanned Aircraft Systems (ICUAS)* (May 2014), pp. 750–755.

[38] FABRESSE, F. R., CABALLERO, F., MAZA, I., AND OLLERO, A. Robust Range-Only SLAM for Unmanned Aerial Systems. *J Intell Robot Syst* (Dec. 2015), 1–14.

[39] FABRESSE, F. R., CABALLERO, F., MERINO, L., AND OLLERO, A. Active perception for 3d range-only simultaneous localization and mapping with uavs. In *2016 International Conference on Unmanned Aircraft Systems (ICUAS)* (June 2016).

[40] FABRESSE, F. R., CABALLERO, F., AND OLLERO, A. An efficient approach for undelayed range-only SLAM based on gaussian mixtures expectation. *Under review at Robotics, IEEE Transactions on*.

[41] FABRESSE, F. R., CABALLERO, F., AND OLLERO, A. Indoor And Outdoor Aerial Robot Datasets Using Range-Only Sensors. *Under review at Journal of Field Robotics*.

[42] FABRESSE, F. R., CABALLERO, F., AND OLLERO, A. Novel multi-hypotheses approach for 3d range-only SLAM using aerial vehicles. In *ICRA Workshop on Aerial Robots Physically Interacting With The Environment* (2014).

[43] FABRESSE, F. R., CABALLERO, F., AND OLLERO, A. Decentralized simultaneous localization and mapping for multiple aerial vehicles using range-only sensors. In *2015 IEEE International Conference on Robotics and Automation (ICRA)* (May 2015), pp. 6408–6414.

[44] FABRESSE, F. R., VÁZQUEZ, A., AND CABALLERO, F. Nanotron radio-based range-only driver package for ros framework. `http://wiki.ros.org/nanotron_swarm`.

[45] FERNÁNDEZ, J. C., MARTINEZ-DE DIOS, J. R., MAZA, I., FABRESSE, F. R., AND OLLERO, A. Ten Years of Cooperation Between Mobile Robots and Sensor Networks. *International Journal of Advanced Robotic Systems* (June 2015).

[46] GASPARRI, A., AND PASCUCCI, F. An interlaced extended information filter for self-localization in sensor networks. *IEEE Transactions on Mobile Computing 9*, 10 (2010), 1491–1504.

[47] GHOLAMI, M., CAI, N., AND BRENNAN, R. An artificial neural network approach to the problem of wireless sensors network localization. *Robotics and Computer-Integrated Manufacturing 29*, 1 (Feb. 2013), 96–109.

[48] GMBH, N. T. Real time location systems (RTLS), apr 2007.

[49] GONZÁLEZ, J., BLANCO, J., GALINDO, C., ORTIZ-DE GALISTEO, A., FERNÁNDEZ-MADRIGAL, J., MORENO, F., AND MARTÍNEZ, J. Mobile robot localization based on ultra-wide-band ranging: A particle filter approach. *Robotics and Autonomous Systems 57*, 5 (May 2009), 496–507.

[50] GORDON, N., SALMOND, D., AND SMITH, A. Novel approach to nonlinear/non-gaussian bayesian state estimation. *IEE Proceedings F Radar and Signal Processing 140*, 2 (1993), 107.

[51] GUOYU FU, JIN ZHANG, WENYUAN CHEN, FENGCHAO PENG, PEI YANG, AND CHUNLIN CHEN. Precise localization of mobile robots via odometry and wireless sensor network. *Int J Adv Robotic Sy 10* (Feb. 2013).

[52] HAI, D., LI, Y., ZHANG, H., AND LI, X. Simultaneous localization and mapping of robot in wireless sensor network. In *2010 IEEE International Conference on Intelligent Computing and Intelligent Systems (ICIS)* (Oct. 2010), vol. 3, pp. 173 –178.

[53] HARA, S., ANZAI, D., YABU, T., DERHAM, T., AND ZEMEK, R. Analysis on TOA and TDOA location estimation performances in a cellular system. In *2011 IEEE International Conference on Communications (ICC)* (June 2011), IEEE, pp. 1–5.

[54] HENRY, P., KRAININ, M., HERBST, E., REN, X., AND FOX, D. RGB-D mapping: Using kinect-style depth cameras for dense 3D modeling of indoor environments. *International Journal of Robotics Research (IJRR) 31*, 5 (April 2012), 647–663.

[55] HERRANZ, F., AND OCAÃŚA, M. WiFi Range-only Data for Localization, Dec. 2013.

[56] HERRERO, D., AND MARTÍNEZ, H. Range-only fuzzy voronoi-enhanced localization of mobile robots in wireless sensor networks. *Robotica FirstView* (2011), 1–15.

[57] HERSHEY, J., AND OLSEN, P. Approximating the kullback leibler divergence between gaussian mixture models. In *IEEE International Conference on Acoustics, Speech and Signal Processing, 2007. ICASSP 2007* (Apr. 2007), vol. 4, pp. IV–317 –IV–320.

[58] HUANG, A. S., BACHRACH, A., HENRY, P., KRAININ, M., FOX, D., AND ROY, N. Visual odometry and mapping for autonomous flight using an rgb-d camera. In *In Proc. of the Intl. Sym. of Robot. Research* (2011).

[59] HUBER, M. F., BAILEY, T., DURRANT-WHYTE, H., AND HANEBECK, U. D. On entropy approximation for gaussian mixture random vectors. In *Multisensor Fusion and Integration for Intelligent Systems, MFI. IEEE International Conference on* (2008), pp. 181–188.

[60] INGWER, B., AND GROENEN, P. J. F. *Modern Multidimensional Scaling - Theory and Applications*. Springer, New York, NY, 1997.

[61] JIANG, Y., AND LEUNG, V. C. An asymmetric double sided two-way ranging for crystal offset. In *International Symposium on Signals, Systems and Electronics, 2007. ISSSE '07* (July 2007), IEEE, pp. 525–528.

[62] JULIER, S. J., AND UHLMANN, J. K. New extension of the kalman filter to nonlinear systems. *Signal Processing, Sensor Fusion, and Target Recognition VI* (July 1997), 182–193.

[63] JULIER, S. J., AND UHLMANN, J. K. Using covariance intersection for SLAM. *Robotics and Autonomous Systems 55*, 1 (Jan. 2007), 3–20.

[64] KLINGBEIL, L., AND WARK, T. A wireless sensor network for real-time indoor localisation and motion monitoring. In *International Conference on Information Processing in Sensor Networks, 2008. IPSN '08* (Apr. 2008), pp. 39 –50.

[65] KOHLBRECHER, S., MEYER, J., VON STRYK, O., AND KLINGAUF, U. A flexible and scalable SLAM system with full 3d motion estimation. In *Proc. IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)* (November 2011), IEEE.

[66] KURT-YAVUZ, Z., AND YAVUZ, S. A comparison of EKF, UKF, FastSLAM2.0, and UKF-based FastSLAM algorithms. In *2012 IEEE 16th International Conference on Intelligent Engineering Systems (INES)* (June 2012), pp. 37 –43.

[67] LEMAIRE, T., LACROIX, S., AND SOLÀ, J. A practical 3d bearing-only SLAM algorithm. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2005. (IROS 2005)* (2005), pp. 2449–2454.

[68] LI, J., CHENG, L., WU, H., XIONG, L., AND WANG, D. An overview of the simultaneous localization and mapping on mobile robot. In *2012 Proceedings of International Conference on Modelling, Identification Control (ICMIC)* (June 2012), pp. 358 –364.

[69] LI, S., WANG, X., ZHAO, S., WANG, J., AND LI, L. Local semidefinite programming-based node localization system for wireless sensor network applications. *IEEE Systems Journal Early Access Online* (2013).

[70] LIM, H., SINHA, S. N., COHEN, M. F., AND UYTTENDAELE, M. Real-time image-based 6-dof localization in large-scale environments. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2012)* (June 2012).

[71] LÓPEZ, Y. A., GÓMEZ, M. E. D. C., ÁLVAREZ, J. L., AND ANDRÉS, F. L.-H. Evaluation of an RSS-based indoor location system. *Sensors and Actuators A: Physical 167*, 1 (May 2011), 110–116.

[72] MARTINEZ-DE DIOS, J. R., JIMENEZ-GONZALEZ, A., DE SAN BERNABE, A., AND OLLERO, A. *A Remote Integrated Testbed for Cooperating Objects*. SpringerBriefs in Electrical and Computer Engineering. Springer International Publishing, Cham, 2014.

[73] MAYBECK, P. S. *Stochastic Models, Estimation, and Control*. Academic Press, Aug. 1982.

[74] MENEGATTI, E., DANIELETTO, M., MINA, M., PRETTO, A., BARDELLA, A., ZANCONATO, S., ZANUTTIGH, P., AND ZANELLA, A. Autonomous discovery, localization and recognition of smart objects through WSN and image features. In *2010 IEEE GLOBECOM Workshops (GC Wkshps)* (Dec. 2010), pp. 1653 –1657.

[75] MERINO, L., CABALLERO, F., AND OLLERO, A. Active sensing for range-only mapping using multiple hypothesis. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on* (2010).

[76] MOORE, D., LEONARD, J., RUS, D., AND TELLER, S. Robust distributed network localization with noisy range measurements. In *Proceedings of the 2Nd International Conference on Embedded Networked Sensor Systems* (New York, NY, USA, 2004), SenSys '04, ACM, pp. 50–61.

[77] NANOTRON TECHNOLOGIES GMBH. nanoPAN development kit, 2013.

[78] NEDDERMEYER, J. C. Nonparametric particle filtering and smoothing with quasi-monte carlo sampling. *Journal of Statistical Computation and Simulation 81*, 11 (2011), 1361–1379.

[79] O. HERO III, ALFRED, PATWARI, NEAL, AND SRICHARAN, KUMAR. CRAWDAD dataset umich/rss (v. 2011-08-10), Aug. 2011.

[80] OLSON, E., LEONARD, J., AND TELLER, S. Robust range-only beacon localization. In *In Proceedings of Autonomous Underwater Vehicles* (2004), pp. 66–75.

[81] PARASURAMAN, RAMVIYAS, CACCAMO, SERGIO, BABERG, FREDRIK, AND OGREN, PETTER. CRAWDAD dataset kth/rss (v. 2016-01-05), Jan. 2016.

[82] PAZ, L. M., PINIÉS, P., TARDÓS, J. D., AND NEIRA, J. Large-scale 6-dof SLAM with stereo-in-hand. *Robotics, IEEE Transactions on 24*, 5 (2008), 946–957.

[83] PEREZ-GRAU, F. J., FABRESSE, F. R., CABALLERO, F., VIGURIA, A., AND OLLERO, A. Long-term aerial robot localization based on visual odometry and radio-based ranging. In *2016 International Conference on Unmanned Aircraft Systems (ICUAS)* (June 2016).

[84] PHILLIPS, CALEB, AND W. ANDERSON, ERIC. CRAWDAD dataset cu/cu_wart (v. 2011-10-24), Oct. 2011.

[85] PRIYANTHA, N. B., CHAKRABORTY, A., AND BALAKRISHNAN, H. The cricket location-support system. In *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking* (New York, NY, USA, 2000), MobiCom '00, ACM, pp. 32–43.

[86] QUIGLEY, M., CONLEY, K., GERKEY, B. P., FAUST, J., FOOTE, T., LEIBS, J., WHEELER, R., AND NG, A. Y. Ros: an open-source robot operating system. In *ICRA Workshop on Open Source Software* (2009).

[87] ROSENBLATT, J. *DAMN: A Distributed Architecture for Mobile Navigation*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, January 1997.

[88] RUNNALLS, A. Kullback-leibler approach to gaussian mixture reduction. *IEEE Transactions on Aerospace and Electronic Systems 43*, 3 (July 2007), 989 –999.

[89] SOLÀ, J., MONIN, A., DEVY, M., AND LEMAIRE, T. Undelayed initialization in bearing only SLAM. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2005. (IROS 2005)* (2005), pp. 2499–2504.

[90] STACHNISS, C., GRISETTI, G., AND BURGARD, W. Information-gain based exploration using rao-blackwellized particle filters. In *Proceedins of Robotics: Sciene and Systems Conference* (2005).

[91] STUMP, E., KUMAR, V., GROCHOLSKY, B., AND SHIROMA, P. Control for localization of targets using range-only sensors. *International Journal of Robotics Research 28*, 6 (2009), 743–757.

[92] THRUN, S., BURGARD, W., AND FOX, D. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2001.

[93] TORRES-GONZÁLEZ, A., DIOS, J. R., AND OLLERO, A. Efficient robot-sensor network distributed SEIF range-only SLAM. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on* (2014), IEEE, pp. 1319–1326.

[94] TORRES-GONZÁLEZ, A., DIOS, J. R. M.-D., AND OLLERO, A. Exploiting Multi-hop Inter-beacon Measurements in RO-SLAM. *Procedia Computer Science 32* (2014), 1101–1107.

[95] TORRES-GONZÁLEZ, A., MARTINEZ-DE DIOS, J. R., AND OLLERO, A. An Adaptive Scheme for Robot Localization and Mapping with Dynamically Configurable Inter-Beacon Range Measurements. *Sensors 14*, 5 (Apr. 2014), 7684–7710.

[96] TORRES-GONZÁLEZ, A., MARTINEZ-DE DIOS, J. R., AND OLLERO, A. Integrating Internode Measurements in Sum of Gaussians Range Only SLAM. In *ROBOT2013: First Iberian Robotics Conference*. Springer International Publishing, 2014, pp. 473–487.

[97] TORRES-GONZÁLEZ, A., MARTINEZ-DE DIOS, J. R., AND OLLERO, A. Robot-WSN Cooperation for Scalable Simultaneous Localization and Mapping. In *Cooperative Robots and Sensor Networks 2014*. Springer Berlin Heidelberg, 2014, pp. 25–41.

[98] TORRES-GONZALEZ, A., MARTINEZ-DE DIOS, J. R., AND OLLERO, A. Accurate fast-mapping Range-Only SLAM for UAS applications. In *Unmanned Aircraft Systems (ICUAS), 2015 International Conference on* (2015), IEEE, pp. 543–550.

[99] TUPYSEV, V. A generalized approach to the problem of distributed Kalman filtering. In *Guidance, Navigation, and Control Conference and Exhibit*. American Institute of Aeronautics and Astronautics, 1998.

[100] VALLICROSA, G., RIDAO, P., AND RIBAS, D. AUV Single Beacon Range-Only SLAM with a SOG Filter. *IFAC-PapersOnLine 48*, 2 (2015), 26–31.

[101] VALLICROSA, G., RIDAO, P., RIBAS, D., AND PALOMER, A. Active Range-Only beacon localization for AUV homing. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on* (2014), IEEE, pp. 2286–2291.

[102] W. ANDERSON, ERIC, AND PHILLIPS, CALEB. CRAWDAD dataset cu/antenna (v. 2009-05-08), 2009.

[103] WANG, H., WAN, J., AND LIU, R. A novel ranging method based on RSSI. *Energy Procedia 12* (2011), 230–235.

[104] WANG, Z. M., MIAO, D. H., AND DU, Z. J. Simultaneous localization and mapping for mobile robot based on an improved particle filter algorithm. In *International Conference on Mechatronics and Automation, 2009. ICMA 2009* (Aug. 2009), pp. 1106 –1110.

[105] XU, E., DING, Z., AND DASGUPTA, S. Robust and Low Complexity Source Localization in Wireless Sensor Networks Using Time Difference of Arrival Measurement. In *2010 IEEE Wireless Communications and Networking Conference (WCNC)* (Apr. 2010), pp. 1–5.

[106] YANG, P. Efficient particle filter algorithm for ultrasonic sensor-based 2d range-only simultaneous localisation and mapping application. *IET Wireless Sensor Systems 2*, 4 (Dec. 2012), 394 –401.

[107] ZHENG, J., WU, C., CHU, H., AND XU, Y. An improved RSSI measurement in wireless sensor networks. *Procedia Engineering 15* (2011), 876–880.

[108] ZHOU, K., AND ROUMELIOTIS, S. Optimal motion strategies for range-only constrained multisensor target tracking. *IEEE Transactions on Robotics 24* (2008), 1168–1185.